



TESIS - TE142599

# **STEGANOGRAFI TEKS MENGGUNAKAN PEMETAAN DIGIT BINER PADA KARAKTER ASCII UNTUK KEAMANAN PLAIN TEXT**

ALFIN NAHARUDDIN  
NRP 07111650067006

DOSEN PEMBIMBING  
Dr. Surya Sumpeno, S.T., M.Sc.  
Dr. Adhi Dharma Wibawa, S.T., M.T.

PROGRAM MAGISTER  
BIDANG KEAHLIAN TELEMATIKA  
DEPARTEMEN TEKNIK ELEKTRO  
FAKULTAS TEKNOLOGI ELEKTRO  
INSTITUT TEKNOLOGI SEPULUH NOPEMBER  
SURABAYA  
2018



TESIS - TE142599

# **STEGANOGRAFI TEKS MENGGUNAKAN PEMETAAN DIGIT BINER PADA KARAKTER ASCII UNTUK KEAMANAN PLAIN TEXT**

ALFIN NAHARUDDIN  
NRP 07111650067006

DOSEN PEMBIMBING  
Dr. Surya Sumpeno, S.T., M.Sc.  
Dr. Adhi Dharma Wibawa, S.T., M.T.

PROGRAM MAGISTER  
BIDANG KEAHLIAN TELEMATIKA  
DEPARTEMEN TEKNIK ELEKTRO  
FAKULTAS TEKNOLOGI ELEKTRO  
INSTITUT TEKNOLOGI SEPULUH NOPEMBER  
SURABAYA  
2018

## LEMBAR PENGESAHAN

Tesis disusun untuk memenuhi salah satu syarat memperoleh gelar  
Magister Teknik (M.T)

di  
Institut Teknologi Sepuluh Nopember

oleh:

Alfin Naharuddin  
NRP. 07111650067006

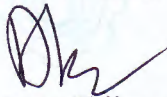
Tanggal Ujian : 6 Juli 2018  
Periode Wisuda : September 2018

Disetujui oleh:



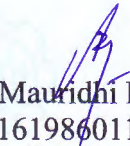
1. Dr. Surya Sumpeno, S.T., M.Sc.  
NIP 196906131997021003

(Pembimbing I)



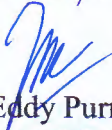
2. Dr. Adhi Dharma Wibawa, S.T., M.T.  
NIP 197605052008121003

(Pembimbing II)



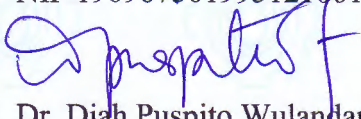
3. Prof. Dr. Ir. Mauridhi Hery Purnomo, M.Eng.  
NIP 195809161986011001

(Penguji)



4. Dr. I Ketut Eddy Purnama, S.T., M.T.  
NIP 196907301995121001

(Penguji)



5. Dr. Diah Puspito Wulandari, S.T., M.Sc.  
NIP 198012192005012001

(Penguji)



Dekan Fakultas Teknologi Elektro

Dr. Tri Arief Sardjono, S.T., M.T.  
NIP. 197002121995121001

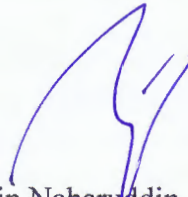
*Halaman ini sengaja dikosongkan*

## PERNYATAAN KEASLIAN TESIS

Dengan ini saya menyatakan bahwa isi keseluruhan Tesis saya dengan judul **“STEGANOGRAFI TEKS MENGGUNAKAN PEMETAAN DIGIT BINER PADA KARAKTER ASCII UNTUK KEAMANAN PLAIN TEXT”** adalah benar-benar hasil karya intelektual mandiri, diselesaikan tanpa menggunakan bahan-bahan yang tidak diijinkan dan bukan merupakan karya pihak lain yang saya akui sebagai karya sendiri.

Semua referensi yang dikutip maupun dirujuk telah ditulis secara lengkap pada daftar pustaka. Apabila ternyata pernyataan ini tidak benar, saya bersedia menerima sanksi sesuai peraturan yang berlaku.

Surabaya, Mei 2018



Alfin Naharuddin

NRP. 07111650067006

*Halaman ini sengaja dikosongkan*

# STEGANOGRAFI TEKS MENGGUNAKAN PEMETAAN DIGIT BINER PADA KARAKTER ASCII UNTUK KEAMANAN PLAIN TEXT

Nama mahasiswa : Alfin Naharuddin  
NRP : 07111650067006  
Pembimbing : 1. Dr. Surya Sumpeno, S.T., M.Sc.  
2. Dr. Adhi Dharma Wibawa, S.T., M.T.

## ABSTRAK

Sebagian besar berita online dan media sosial menggunakan teks biasa (*plain text*) sebagai media penyampaian informasi, karena bersifat ringan dan multi-platform. Namun *plain text* juga mudah untuk diedit ulang dan didistribusikan kembali, sehingga ada kesempatan bagi pihak luar untuk mengubah teks dan mendistorsi maksud dan tujuan dari informasi yang sebenarnya. Kondisi ini semakin rentan jika informasi tersebut tidak menyebutkan sumber berita atau tautan yang valid. Salah satu strategi untuk mengatasi masalah ini adalah dengan menyisipkan teks rahasia kedalam keseluruhan *plain text* tersebut sehingga setiap perubahan mengenai informasi teks dapat dideteksi. Penelitian ini mengusulkan metode penyisipan teks rahasia ke dalam teks sampul dengan cara memetakan digit biner (bit) teks rahasia ke digit biner karakter teks sampul berbasis karakter *American Standart Code for Information Interchange* (ASCII). Pemetaan bit pada teks sampul ini melibatkan semua karakter termasuk spasi, tanda baca, dan simbol. Dalam pra-pemrosesan, teks rahasia dienkripsi menggunakan *One Time Pad* (OTP) ke dalam *ciphertext*, dan setiap karakter diubah menjadi nilai biner sepanjang 7 bit, begitu juga pada setiap karakter teks sampul juga diubah menjadi nilai biner 7 bit. Proses *embedding* dilakukan dengan memetakan satu bit teks rahasia ke bit pertama dari karakter teks sampul yang memiliki nilai bit yang sama (mis. Bit 1 dari karakter teks rahasia ke bit pertama 1 dari karakter teks sampul). Proses serupa dilakukan untuk bit 0 teks rahasia. Proses ini diulang sampai semua bit teks rahasia masuk ke seluruh teks sampul. Setiap kali satu bit teks rahasia mendapat posisi mereka dalam satu bit teks sampul maka sistem mencatat dari setiap posisi bit sebagai kunci stego. Nantinya kunci stego ini juga berfungsi sebagai kunci untuk mengekstrak teks rahasia yang telah disisipkan ke teks sampul. Hasil proses penyisipan ini tidak mengubah tampilan teks sampul sehingga metode ini bekerja secara optimal untuk menyembunyikan informasi rahasia yang dapat berfungsi sebagai steganografi teks atau sebagai *watermark*. Dapat disimpulkan bahwa semua karakter teks sampul dapat digunakan sebagai media penyembunyian teks rahasia di mana 1 karakter teks rahasia membutuhkan 7 karakter teks sampul. Selain itu, teks stego yang dihasilkan terlihat sama persis dengan teks sampul, hal ini dibuktikan dari hasil pengukuran dengan *similarity distance* menggunakan *Jaro-Winkler Distance* yang menghasilkan nilai 1, dan setelah disertakan *identifier* nilai *Jaro Distance* bernilai 0,9972 dan *Jaro-Winkler Distance* bernilai 0,9983.

Kata kunci: steganografi teks, pemetaan bit, plain text, karakter ASCII.

*Halaman ini sengaja dikosongkan*



# TEXT STEGANOGRAPHY USING BINARY DIGIT MAPPINGS ON ASCII CHARACTERS FOR PLAIN TEXT SECURITY

By : Alfin Naharuddin  
Student Identity Number : 07111650067006  
Supervisors : 1. Dr. Surya Sumpeno, S.T., M.Sc.  
2. Dr. Adhi Dharma Wibawa, S.T., M.T.

## ABSTRACT

Most of online news and social media use plain text as a medium for submission of information, because it is lightweight and multi-platform. However plain text is also easy to re-edit and redistributed, so that there is an opportunity for outside parties to change the text and distort the intent and purpose of the actual information. This condition will make it even more vulnerable if the information is not included in news source or a valid link. One strategy to overcome this problem is by embedding a secret text into the whole plain text of the information so that any change regarding the text information then can be detected. This paper proposes a method of embedding a Secret-Text into a Cover-Text by mapping the binary digit (bit) of each *American Standart Code for Information Interchange* (ASCII) character of Secret-Text into the binary digit of each of cover-text character. For mapping purpose, all the binary digit of cover-text including spaces, punctuation, and symbols were involved. In the pre-processing, the secret-text was encrypted using One Time Pad (OTP) into a ciphertext, and each character was converted into a binary value of 7 bits. Meanwhile, each cover-text character was also converted into a binary value of 7 bits. The embedding process was done by mapping a single bit of Secret-Text into the first bit of cover-text character that has the same bit value (eg. bit 1 of secret-text character to the 1st bit 1 of cover-text character). Similar process was done for bit 0 of secret-text. This process was repeated until all bits of Secret-Text entered into the entire cover text. Each time one bit of secret-text got their position in one bit of cover-text, the system then created a chain code of each bit's position as a stegokey. Later on this stegokey also serves as a key to extract the Secret-Text that has been embedded to the cover-text. This embedding process result did not change the appearance of the cover text so that this method works optimally for the concealment of confidential information that can be functioned as text steganography or as a watermark. We conclude that all cover-text characters can be used as secret-text concealment media where 1 character of Secret Text needs 7 characters of cover-text. Moreover, the stegotext we used looks exactly the same as Cover-Text, this result was showed by the value of *similarity distance* test using *Jaro-Winkler Distance* that yielded 1.0 value, and after included the identifier, *Jaro Distance* yielded 0.9972 value and *Jaro-Winkler Distance* yielded 0.9983 value.

Key words: text steganography, bit mapping, plain text, ASCII characters.

*Halaman ini sengaja dikosongkan*

## KATA PENGANTAR

*Alhamdulillah Rabbal Alamin*, Puji syukur penulis panjatkan kepada Allah SWT. atas rahmat dan kasih karunia-Nya dan juga Shalawat serta Salam untuk Muhammad Rasul junjungan sekalian alam yang kebaikan hatinya menyejukkan umat segala bangsa, dengannya tesis ini dapat selesai dengan baik dan lancar. Terima kasih kepada Ayah (alm), Ibu, Mertua, Istri, Anak, Adik, Ipar dan semua pihak, baik yang disebut atau tidak dalam tesis ini, yang telah memberikan dukungan moril maupun materil demi selesainya tesis ini.

Ucapan terimakasih dan penghargaan penulis sampaikan kepada Dr. Surya Sumpeno, S.T., M.Sc. selaku pembimbing Utama dan Dr. Adhi Dharma Wibawa, S.T., M.T. selaku pembimbing kedua yang juga selaku Koordinator Bidang Keahlian Telematika, yang dengan penuh perhatian, dan berdedikasi meluangkan waktu, memberikan pengarahan, dan motivasi serta semangat dalam penulisan tesis ini. Penulis dapat menyelesaikan tesis ini tidak terlepas dari bantuan, kerjasama dan dukungan dari berbagai pihak. Untuk itu penulis menyampaikan terimakasih kepada:

1. Kementerian Agama Republik Indonesia yang telah memberi kesempatan untuk meneruskan studi kejenjang yang lebih tinggi;
2. Kementerian Komunikasi dan Informatika Republik Indonesia yang telah membuka peluang belajar dan memberi dukungan berupa beasiswa;
3. Institut Teknologi Sepuluh Nopember Surabaya atas kesempatan, dukungan, dan fasilitas untuk bisa menempuh jenjang magister;
4. Prof. Ir. Joni Hermana, M.Sc.Es, Ph.D. selaku Rektor Institut Teknologi Sepuluh Nopember Surabaya;
5. Dr. Tri Arief Sardjono, ST., MT. selaku Dekan Fakultas Teknologi Elektro, Institut Teknologi Sepuluh Nopember Surabaya;
6. Dr. Eng. Ardyono Priyadi, S.T., M.Eng. selaku Kepala Departemen Teknik Elektro, Fakultas Teknologi Elektro, Institut Teknologi Sepuluh Nopember Surabaya;

7. Dr. Ir. Wirawan, DEA selaku Kepala Program Studi Pascasarjana Departemen Teknik Elektro, Fakultas Teknologi Elektro, Institut Teknologi Sepuluh Nopember Surabaya;
8. Eko Setijadi, S.T., M.T., Ph.D. selaku Dosen Wali Akademik untuk mahasiswa Pengelola TIK Pemerintahan (PeTIK) 2016;
9. Seluruh Dosen dan staf Program Studi Magister (S2) Departemen Teknik Elektro, khususnya Bidang Keahlian Telematika/Pengelola TIK Pemerintahan (PeTIK), atas jasa dan pengabdianya;
10. Rekan-Rekan PeTIK 2016 serta Telematika 2016 atas semua perjalanan dilalui bersama, semoga selalu terjalin silaturahmi dan terus bersinergi;
11. Serta seluruh pihak yang tidak dapat penulis sebutkan satu persatu

Surabaya, 3 Mei 2018

Penulis

## DAFTAR ISI

LEMBAR PENGESAHAN .....	iii
PERNYATAAN KEASLIAN TESIS .....	v
ABSTRAK .....	vii
ABSTRACT .....	ix
KATA PENGANTAR .....	xi
DAFTAR ISI .....	xiii
DAFTAR GAMBAR .....	xvii
DAFTAR TABEL .....	xix
BAB 1 PENDAHULUAN .....	1
1.1 Latar Belakang .....	1
1.2 Rumusan Masalah .....	2
1.3 Tujuan .....	3
1.4 Batasan Masalah .....	3
1.5 Kontribusi .....	3
1.6 Metodologi Penelitian .....	3
BAB 2 KAJIAN PUSTAKA .....	5
2.1 Kajian Penelitian Terkait .....	5
2.2 Teori Dasar .....	7
2.2.1 Otentikasi .....	7
2.2.2 Plain Text (Teks Biasa) .....	8
2.2.3 ASCII .....	8
2.2.4 Enkripsi One Time Pad .....	15
2.2.5 Steganografi .....	16
2.2.6 Kapasitas .....	23
2.2.7 Jaro-Winkler Distance .....	24
2.2.8 Optical Character Recognition (OCR) .....	25
BAB 3 METODOLOGI PENELITIAN .....	27

3.1	Skema Penelitian .....	27
3.1.1	Identifikasi Masalah dan Motivasi .....	28
3.1.2	Penentuan Tujuan Penelitian .....	28
3.1.3	Perancangan dan Pengembangan Solusi .....	28
3.1.4	Demonstrasi .....	29
3.1.5	Evaluasi .....	29
3.1.6	Komunikasi.....	29
3.2	Analisis Sistem.....	29
3.3	Perancangan Sistem.....	32
3.3.1	Enkripsi dengan One Time Pad (OTP) menggunakan Modulo 26..	32
3.3.2	Dekripsi dengan One Time Pad (OTP) menggunakan Modulo 26 .	33
3.3.3	Konversi ASCII ke Nilai Biner Digit (Bit).....	34
3.3.4	Konversi Nilai Biner Digit (Bit) ke ASCII.....	36
3.3.5	Metode Pemetaan Binary Digit (Bit).....	37
3.3.6	Proses Penyisipan .....	39
3.3.7	Proses Ekstraksi .....	40
3.4	Implementasi Sistem .....	42
3.4.1	Penyisipan Teks Rahasia .....	42
3.4.2	Ekstraksi Teks Rahasia.....	43
3.4.3	Detil Proses Pemetaan Bit .....	44
3.4.4	Uji Similarity .....	44
BAB 4 HASIL DAN PEMBAHASAN .....		47
4.1	Penerapan Metode Pemetaan Bit.....	47
4.1.1	Enkripsi dan Dekripsi OTP Mod 26 .....	47
4.1.2	Penyisipan Teks Rahasia .....	49
4.1.3	Ekstraksi Teks Rahasia.....	53
4.2	Kapasitas Penyisipan Teks Rahasia pada Teks Sampul.....	55
4.3	Imperseptibilitas Teks Stego .....	56
4.4	Penyisipan Identifier.....	57
4.5	Analisis Teks Stego dengan OCR (Optical Character Recognition).....	59
BAB 5 PENUTUP .....		63
5.1	Kesimpulan.....	63

5.2	Saran .....	64
DAFTAR PUSTAKA .....		65
LAMPIRAN .....		69

*Halaman ini sengaja dikosongkan*



## DAFTAR GAMBAR

Gambar 2.1 Proses steganografi .....	17
Gambar 2.2 Metode Steganografi Text .....	18
Gambar 2.3 Metode Steganografi Teks dengan Penyembunyian Paragraf .....	20
Gambar 2.4 Cover File .....	22
Gambar 2.5 Stego File .....	22
Gambar 2.6 Metode pemetaan bit .....	23
Gambar 2.7 Diagram blok OCR .....	25
Gambar 2.8 (a) Pengenalan karakter secara offline, dan (b) secara online .....	26
Gambar 3.1 Skema Penelitian dengan metode DSRM .....	27
Gambar 3.2 Skema steganografi teks dengan metode pemetaan bit .....	30
Gambar 3.3 Diagram alir enkripsi menggunakan OTP mod 26 .....	32
Gambar 3.4 Diagram alir dekripsi menggunakan OTP mod 26 .....	34
Gambar 3.5 Diagram alir konversi karakter ASCII ke dalam nilai binernya .....	35
Gambar 3.6 Diagram alir konversi deret bit kedalam karakter ASCII .....	36
Gambar 3.7 Diagram alir Pemetaan Binary Digit (Bit) .....	38
Gambar 3.8 Diagram alir penyisipan teks rahasia dengan pemetaan bit .....	39
Gambar 3.9 Diagram alir ekstraksi teks rahasia dengan pemetaan bit .....	41
Gambar 3.10 Antar muka Penyisipan Teks Rahasia .....	42
Gambar 3.11 Antar muka Ekstraksi Teks Rahasia .....	43
Gambar 3.12 Antar muka Detil Proses Pemetaan Bit .....	44
Gambar 3.13 Antar muka Uji Similarity Pemetaan Bit .....	45
Gambar 4.1 Format kunci OTP Modulo 26, dengan huruf kapital dan huruf kecil berlaku sama .....	47
Gambar 4.2 Teks Rahasia Visi Kementerian Agama RI .....	48
Gambar 4.3 Teks terenkripsi Visi Kementerian Agama RI .....	48
Gambar 4.4 Kunci OTP yang dibuat terlebih dahulu secara random .....	49
Gambar 4.5 Teks rahasia hasil dekripsi .....	49
Gambar 4.6 Pseudocode penyisipan teks rahasia .....	49
Gambar 4.7 Teks Sampul .....	51
Gambar 4.8 Teks Stego .....	52
Gambar 4.9 Kunci Stego .....	53
Gambar 4.10 Pseudocode ekstraksi teks rahasia .....	55
Gambar 4.11 Teks rahasia hasil ekstraksi .....	55
Gambar 4.12 Teks stego yang telah disertakan identifier .....	58
Gambar 4.13 File 1 .....	60
Gambar 4.14 File 2 .....	61
Gambar 4.15 File 3 .....	61

*Halaman ini sengaja dikosongkan*

## **DAFTAR TABEL**

Tabel 2.1 Daftar penelitian terkait .....	5
Tabel 2.2 Tabel Konversi ASCII .....	9
Tabel 4.1 Hasil pengujian menggunakan OCR (Optical Character Recognition)	59

*Halaman ini sengaja dikosongkan*

# **BAB 1**

## **PENDAHULUAN**

### **1.1 Latar Belakang**

Budaya instan dan serba cepat serta masifnya penggunaan teknologi informasi terutama dalam media sosial dan aplikasi *messaging* membawa dampak tersendiri pada masyarakat yaitu kemudahan berbagi informasi, namun tidak semua masyarakat mempunyai daya nalar yang sama sehingga sebuah informasi yang diterima akan ditanggapi berbeda dan ditindaklanjuti secara beragam pula. Hal ini sering membuat kerancuan tersendiri jika dihadapkan pada sebuah informasi yang berupa *plain text* dimana dengan mudah di *copy-paste* dan disebarkan kembali pada berbagai macam media, dimana tidak mudah untuk mengkonfirmasi keabsahan dan keutuhan informasi yang berupa *plain text* jika tidak disertakan sumber asal atau tautan aslinya.

Berdasarkan Undang-Undang Republik Indonesia Nomor 19 tahun 2016 tentang perubahan atas Undang-Undang Nomor 11 Tahun 2008 [1] tentang Informasi dan Transaksi Elektronik disebutkan bahwa Bahwa Penyelenggara Sistem Elektronik wajib mengoperasikan Sistem Elektronik yang memenuhi persyaratan minimum yaitu dapat melindungi ketersediaan, keutuhan, keotentikan, kerahasiaan, dan keteraksesan Informasi Elektronik. Dimana Pemerintah memfasilitasi pemanfaatan Teknologi Informasi dan Transaksi Elektronik sesuai dengan ketentuan peraturan perundang-undangan dan Pemerintah melindungi kepentingan umum dari segala jenis gangguan sebagai akibat penyalahgunaan Informasi Elektronik dan Transaksi Elektronik yang mengganggu ketertiban umum [2], serta Pemerintah wajib melakukan pencegahan penyebarluasan dan penggunaan Informasi Elektronik dan/atau Dokumen Elektronik yang memiliki muatan yang dilarang sesuai dengan ketentuan peraturan perundang-undangan [3].

Berangkat dari fakta tersebut perlu ada sebuah mekanisme penunjang regulasi untuk menyatakan sebuah informasi tersebut benar adanya, yaitu salah satunya dengan cara menyematkan suatu pesan khusus atau pesan rahasia pada

informasi yang berupa *plain text*, hal ini dapat diimplementasikan dengan menerapkan steganografi teks yang berfungsi sebagai pengaman *plain text* [4].

Metode steganografi teks yang diterapkan pada *plain text* dengan memanfaatkan nilai biner pada karakter *American Standart Code for Information Interchange* (ASCII) memungkinkan hasil yang diperoleh dari hasil penyisipannya tidak mengubah tampilan teks sampul yang digunakan sebagai media penyisipannya. Pada metode steganografi *hiding text in paragraph* [5] pesan rahasia disembunyikan pada setiap kata yang tidak mempunyai kesamaan huruf awal dan akhir dalam sebuah paragraf, metode ini menghasilkan teks stego yang sama persis dengan teks sampul yang digunakan namun kapasitas penyimpanannya terbatas dan tidak bisa berfungsi sebagai pengaman *plain text* karena tidak semua karakter disisipi dengan pesan rahasia. Selanjutnya pada penelitian steganografi image [6] menggunakan pesan rahasia menggunakan skema *Bits mapping using Dot Pattern Character Encoding*, penyisipan pesan rahasia yang dikonversi dalam bentuk angka biner yang didapatkan dari skema pemetaan menggunakan pola titik yang telah ditentukan, kemudian disisipkan ke dalam image dengan metode *Least Significant Bit* (LSB).

Dalam penelitian ini diajukan sebuah metode pemetaan bit pada karakter ASCII, metode ini memanfaatkan nilai kode biner pada karakter ASCII dengan cara menyisipkan pesan rahasia kedalam setiap karakter teks sampul yang digunakan sehingga tidak menyisakan satupun karakter teks sampul bahkan pada spasi, tanda baca maupun simbol standar ASCII, sehingga selain berfungsi sebagai penampung pesan rahasia juga berperan sebagai pengaman *plain text* yang berfungsi menjaga keutuhan pesan, dimana setiap karakter teks sampul dapat dikonfirmasi keutuhan dan keasliannya.

## **1.2 Rumusan Masalah**

1. Dari pemaparan latar belakang diketahui bahwa informasi yang berupa *plain text* berpotensi diubah sehingga tidak sesuai dengan sumber informasi aslinya karena belum ada mekanisme otentikasi pada *plain text*;

2. Implementasi otentikasi pada plain text (teks biasa) dapat dilakukan dengan cara menyisipkan sebuah pesan rahasia ke dalam teks informasi tersebut yang berfungsi sebagai pengaman teks.

### **1.3 Tujuan**

1. Penelitian ini bertujuan melakukan penyisipan teks rahasia menggunakan metode pemetaan bit pada karakter ASCII, dengan cara mengenkripsi teks rahasia terlebih dahulu sebelum disisipkan, kemudian dikonversikan kedalam nilai binernya, yang diterapkan dengan skema steganografi teks;
2. Dengan plain text yang telah disisipi teks rahasia, maka plain text dapat menjadi informasi yang dapat dikonfirmasi keabsahannya sekaligus bisa menjadi penyampai pesan rahasia.

### **1.4 Batasan Masalah**

Lingkup penelitian ini pada penerapan steganografi teks berupa plain text dengan memanfaatkan karakter standar ASCII, dan dengan menggunakan metode enkripsi simetris *One Time Pad* (OTP).

### **1.5 Kontribusi**

Pemanfaatan penelitian ini dapat diterapkan pada berbagai media digital yang menggunakan plain text sebagai basisnya, misal pada media berita online ataupun *e-commerce* maupun pada dokumen negara. Dengan menggunakan metode bit mapping pada karakter ASCII, setiap karakter pada plain text yang telah disisipi pesan rahasia menjadi media digital yang *imperceptible* serta berfungsi sebagai pengaman plain text yang memungkinkan untuk dikonfirmasi keaslian dan keutuhannya.

### **1.6 Metodologi Penelitian**

Tahapan dan langkah-langkah dalam melakukan penelitian ini meliputi berbagai hal dan kegiatan antara lain:

1. Studi literatur dan pengumpulan informasi

Pada tahapan ini mempelajari dan mencatat berbagai literatur mulai dari buku, jurnal internasional, karya ilmiah (*conference paper*), tugas akhir hingga media online yang membahas teknik penyisipan pesan dan steganografi secara umum serta secara khusus yang membahas steganografi teks dengan metode yang berfokus pada penggunaan nilai kode biner pada karakter ASCII.

## 2. Perancangan sistem

Setelah terkumpul data dan informasi terkait, maka dimulailah perancangan sistem dalam bentuk alur *flowchart*.

## 3. Implementasi

Implementasi dari teori yang telah dirumuskan dituangkan dalam bentuk aplikasi yang berbasis web dengan menggunakan pemrograman *Hypertext Preprocessor* PHP, aplikasi ini mengenkripsi inputan plaintext berupa pesan rahasia menjadi *cipher text* yang kemudian diubah menjadi bentuk kode biner untuk kemudian disematkan ke dalam setiap karakter teks sampul yang berupa plain text.

## 4. Pengujian

Dengan skema steganografi text, metode pemetaan bit pada karakter ASCII diterapkan kemudian untuk mengetahui kapasitas teks sampul mampu menampung pesan rahasia digunakan rumusan,  $kapasitas = \text{teks rahasia} / \text{teks sampul}$ , kemudian untuk menguji kemiripan antara teks sampul dan teks stego digunakan *Jaro-Winkler Distance*.



## BAB 2

### KAJIAN PUSTAKA

Pada bab ini dijelaskan berbagai teori yang menjadi landasan dan acuan penelitian yang dilakukan. Diantara pemaparan yang disampaikan adalah berkaitan dengan kajian penelitian terkait dan teori yang mendasari penelitian serta teori untuk pengujian hasil penelitian.

#### 2.1 Kajian Penelitian Terkait

Beberapa penelitian yang sejenis sebelumnya telah dilakukan, penelitian tersebut membahas tentang steganografi teks berbasis plain text dengan memanfaatkan nilai binary digit pada karakter ASCII namun dengan kapasitas penyisipan pesan rahasia yang terbatas serta tidak memaksimalkan keseluruhan teks sampul yang digunakan. Sedangkan penelitian yang dilakukan ini memaksimalkan teks sampul sebagai media penyisipan pesan rahasia dengan menggunakan nilai *Binary Digit* (bit) pada setiap karakter teks sampul tanpa mengubah tampilannya, sehingga penelitian ini menjadi penting karena selain hasil yang diperoleh mempunyai daya tampung penyisipan rahasia yang besar juga teks sampul tersebut menjadi *secure text* dimana jika dilakukan perubahan pada teks tersebut maka akan dapat diketahui. Berikut beberapa bahasan penelitian yang mendukung tema penelitian ini:

Tabel 2.1 Daftar penelitian terkait

No.	Judul	Jml. Sampel	Variabel	Metode	Validasi	Hasil
1.	Text Steganographic Approaches: a Comparison; Peneliti: Monika Agarwal; Department of Computer Science and Engineering, PDPM-IIITDM, Jabalpur, India; International Journal of Network Security & Its Applications, 2013.	3 Metode steganografi text	Huruf, Daftar Kata, Paragraf, Jumlah Byte dan Ukuran Teks sampul	Missing Letter Puzzle, Hiding Data in Wordlist, Hiding Data in Paragraphs	Rasio kapasitas, Similarity Measure dengan Jaro Winkler Distance	Missing Letter Puzzle, Capacity: 7.833%, Jaro Score: 0.95; Wordlist, Capacity: 7.898%, Jaro Score: 1; Paragraph Approach, Capacity: 2.075%, Jaro Score: 1

No.	Judul	Jml. Sampel	Variabel	Metode	Validasi	Hasil
2.	<p>XORSTEG: A New Model of Text Steganography</p> <p>Peneliti: Tapodhir Acharjee, dkk. Department of Computer Science and Engineering Assam University, Silchar, Assam, India.</p> <p>Communication and Electronics Systems (ICCES), International Conference on 2016.</p>	5 dokumen	Huruf, Kata, Kalimat, Jumlah Byte dan Ukuran Teks sampul	Hiding data in paragraphs with XOR	Rasio Kapasitas	Metode XORSTEG ini lebih efisien karena menampung lebih banyak jumlah bytes dan mempunyai tampilan yang sama antara teks stego dengan cover tex-nya.
3	<p>Secure Digital Steganography for ASCII Text Documents,; Khan Farhan Rafat · M. Sher; Arabian Journal for Science and Engineering, 2013</p>	4 dokumen	Binary digit, spasi	Subtitusi bit pada spasi, Enkripsi XOR	Hash value	Konversi stego object menjadi file .pdf maupun .doc terbukti data tetap utuh dan tidak berubah.
4.	<p>Implementasi Steganografi Pada Media Teks Dengan Metode Line-Shift Coding Dan Metode Centroid,</p> <p>Peneliti: Indri Andiniarti</p> <p>Departemen ilmu komputer</p> <p>Fakultas matematika dan ilmu pengetahuan alam</p> <p>Institut Pertanian Bogor</p> <p>2009</p>	2 dokumen	Baris Font Piksel Bit	Line-shift coding dan Metode Centroid	Ekstraksi dan Resizing	Metode <i>line-shift coding</i> tahan terhadap proses pencetakan, perbedaan kualitas pencetakan, proses fotokopi sampai dua kali fotokopian, dan operasi <i>resizing</i> gambar sampai ukuran yang lebih besar atau sama dengan 700*963 piksel.

No.	Judul	Jml. Sampel	Variabel	Metode	Validasi	Hasil
5.	Identifikasi Watermark Pada Teks Dengan Metode Centroid  Peneliti: Istighfarini  Teknik Elektro Institut Teknologi Sepuluh Nopember  2004	10 Dokumen	Baris Font Piksel Bit	Watermarking Line-shift coding dan Metode Centroid	Ekstraksi dan Resizing	Adanya nilai pergeseran piksel yang ditambahkan saat watermarking dapat merubah nilai centroid, Identifikasi watermark dengan metode centroid membedakan jumlah piksel antara teks asli dengan teks ter-watermark

## 2.2 Teori Dasar

### 2.2.1 Otentikasi

Otentikasi berasal dari bahasa Yunani, *Authentes* yang berarti penulis. Otentikasi merupakan proses pengidentifikasian terhadap seseorang, biasanya berdasarkan pada *username* dan *password*, dengan kata lain otentikasi pada konteks komunikasi komputer, otentikasi dilakukan untuk memastikan siapa pengguna jaringan yang sebenarnya, hal ini penting karena hanya yang mempunyai hak akses saja yang boleh mengakses pada suatu jaringan yang telah ditentukan. Otentikasi memungkinkan administrator jaringan untuk mengidentifikasi siapa yang dapat terkoneksi ke perangkat jaringan atau *resource* tertentu dengan cara memasukkan *username* dan *password*.

Dalam sistem keamanan, otentikasi berbeda dari otorisasi, yang merupakan proses pemberian hak akses terhadap objek-objek sebuah sistem berdasarkan identitas mereka, otentikasi biasanya lebih mengarahkan kepada siapa dirinya dibanding dengan hak akses dari seseorang. Otorisasi didefinisikan sebagai proses mengizinkan atau penolakan mengakses sumber daya jaringan. mekanisme penggunaan resource jaringan dilakukan dalam dua langkah, langkah pertama adalah otentikasi, memeriksa apakah seorang *user* diberi hak akses,

berdasarkan pada *username* dan *password*, tahap dua adalah otorisasi, untuk mengakses berbagai *resource* yang ada untuk melihat apakah dia seorang user yang sah.

Teknik otentikasi (dalam komunikasi data) adalah prosedur yang digunakan untuk membuktikan:

- Keaslian pesan (*Message Integrity*)
- Keaslian identitas pengirim (*User Authentication*)
- Pengirim tidak dapat menyangkal isi pesan (*Non-Repudiation*).

Identifikasi didefinisikan sebagai prosedur atau teknologi yang menghubungkan antara user, group atau entitas lain dengan satu atau beberapa *identifier*.

### 2.2.2 Plain Text (Teks Biasa)

*Plain Text* adalah teks yang mendukung format ASCII. Plain text tidak memiliki format dan informasi struktur seperti ukuran dan tipe *font*, warna, atau *layout*. Plain text biasanya digunakan antar-komputer yang tidak memiliki kesepakatan untuk saling bertukar informasi format dan layout teks. Biasanya plain text berupa ASCII yang setiap karakternya memiliki range antara 7 hingga 8 bit. Contoh penggunaannya : *Notepad* (Windows), *edit* (DOS), *ed*, *emacs*, *vi*, *vim*, *Gedit* atau *nano* (Unix, GNU/Linux), *SimpleText* (Mac OS), atau *TextEdit* (Mac OS X) [7].

Ada perbedaan istilah “plain text” dan “plaintext”, istilah plaintext secara umum digunakan dalam tema kriptografi sebagai teks yang dipakai sebelum proses enkripsi dan hasilnya disebut ciphertext.

### 2.2.3 ASCII

ASCII merupakan kependekan dari kata *American Standard Code for Information Interchange*, dan pengertian dari ASCII sendiri adalah suatu standar internasional dalam kode huruf dan simbol seperti *Hex* dan *Unicode* tetapi ASCII lebih bersifat universal, contohnya 124 adalah untuk karakter "|". Ia selalu digunakan oleh komputer dan alat komunikasi lain untuk menunjukkan teks, sedangkan fungsi dari kode ASCII ialah digunakan untuk mewakili karakter-

karakter angka maupun huruf didalam komputer [8]. Berikut di bawah ini adalah table konversi dari kode ASCII.

Tabel 2.2 Tabel Konversi ASCII [9]

Dec	Hex	Binary	Symb	Char	Description
0	00	00000000		NULL	(Null character)
1	01	00000001		SOH	(Start of Header)
2	02	00000010		STX	(Start of Text)
3	03	00000011		ETX	(End of Text)
4	04	00000100		EOT	(End of Transmission)
5	05	00000101		ENQ	(Enquiry)
6	06	00000110		ACK	(Acknowledgement)
7	07	00000111		BEL	(Bell)
8	08	00001000		BS	(Backspace)
9	09	00001001		HT	(Horizontal Tab)
10	0A	00001010		LF	(Line feed)
11	0B	00001011		VT	(Vertical Tab)
12	0C	00001100		FF	(Form feed)
13	0D	00001101		CR	(Carriage return)
14	0E	00001110		SO	(Shift Out)
15	0F	00001111		SI	(Shift In)
16	10	00010000		DLE	(Data link escape)
17	11	00010001		DC1	(Device control 1)
18	12	00010010		DC2	(Device control 2)
19	13	00010011		DC3	(Device control 3)
20	14	00010100		DC4	(Device control 4)
21	15	00010101		NAK	(Negative acknowledgement)
22	16	00010110		SYN	(Synchronous idle)
23	17	00010111		ETB	(End of transmission block)
24	18	00011000		CAN	(Cancel)
25	19	00011001		EM	(End of medium)
26	1A	00011010		SUB	(Substitute)
27	1B	00011011		ESC	(Escape)
28	1C	00011100		FS	(File separator)
29	1D	00011101		GS	(Group separator)
30	1E	00011110	-	RS	(Record separator)
31	1F	00011111		US	(Unit separator)
32	20	00100000			(Space)
33	21	00100001	!	!	(Exclamation mark)
34	22	00100010	"	"	(Quotation mark ; quotes)

Dec	Hex	Binary	Symb	Char	Description
35	23	00100011	#	#	(Number sign)
36	24	00100100	\$	\$	(Dollar sign)
37	25	00100101	%	%	(Percent sign)
38	26	00100110	&	&	(Ampersand)
39	27	00100111	'	'	(Apostrophe)
40	28	00101000	(	(	(round brackets or parentheses)
41	29	00101001	)	)	(round brackets or parentheses)
42	2A	00101010	*	*	(Asterisk)
43	2B	00101011	+	+	(Plus sign)
44	2C	00101100	,	,	(Comma)
45	2D	00101101	-	-	(Hyphen)
46	2E	00101110	.	.	(Dot , full stop)
47	2F	00101111	/	/	(Slash)
48	30	00110000	0	0	(number zero)
49	31	00110001	1	1	(number one)
50	32	00110010	2	2	(number two)
51	33	00110011	3	3	(number three)
52	34	00110100	4	4	(number four)
53	35	00110101	5	5	(number five)
54	36	00110110	6	6	(number six)
55	37	00110111	7	7	(number seven)
56	38	00111000	8	8	(number eight)
57	39	00111001	9	9	(number nine)
58	3A	00111010	:	:	(Colon)
59	3B	00111011	;	;	(Semicolon)
60	3C	00111100	<	<	(Less-than sign)
61	3D	00111101	=	=	(Equals sign)
62	3E	00111110	>	>	(Greater-than sign ; Inequality)
63	3F	00111111	?	?	(Question mark)
64	40	01000000	@	@	(At sign)
65	41	01000001	A	A	(Capital A)
66	42	01000010	B	B	(Capital B)
67	43	01000011	C	C	(Capital C)
68	44	01000100	D	D	(Capital D)
69	45	01000101	E	E	(Capital E)
70	46	01000110	F	F	(Capital F)
71	47	01000111	G	G	(Capital G)
72	48	01001000	H	H	(Capital H)
73	49	01001001	I	I	(Capital I)
74	4A	01001010	J	J	(Capital J)
75	4B	01001011	K	K	(Capital K)
76	4C	01001100	L	L	(Capital L)

Dec	Hex	Binary	Symb	Char	Description
77	4D	01001101	M	M	(Capital M)
78	4E	01001110	N	N	(Capital N)
79	4F	01001111	O	O	(Capital O)
80	50	01010000	P	P	(Capital P)
81	51	01010001	Q	Q	(Capital Q)
82	52	01010010	R	R	(Capital R)
83	53	01010011	S	S	(Capital S)
84	54	01010100	T	T	(Capital T)
85	55	01010101	U	U	(Capital U)
86	56	01010110	V	V	(Capital V)
87	57	01010111	W	W	(Capital W)
88	58	01011000	X	X	(Capital X)
89	59	01011001	Y	Y	(Capital Y)
90	5A	01011010	Z	Z	(Capital Z)
91	5B	01011011	[	[	(square brackets or box brackets)
92	5C	01011100	\	\	(Backslash)
93	5D	01011101	]	]	(square brackets or box brackets)
94	5E	01011110	^	^	(Caret or circumflex accent)
95	5F	01011111	_	_	(underscore , understrike , underbar or low line)
96	60	01100000	`	`	(Grave accent)
97	61	01100001	a	a	(Lowercase a )
98	62	01100010	b	b	(Lowercase b )
99	63	01100011	c	c	(Lowercase c )
100	64	01100100	d	d	(Lowercase d )
101	65	01100101	e	e	(Lowercase e )
102	66	01100110	f	f	(Lowercase f )
103	67	01100111	g	g	(Lowercase g )
104	68	01101000	h	h	(Lowercase h )
105	69	01101001	i	i	(Lowercase i )
106	6A	01101010	j	j	(Lowercase j )
107	6B	01101011	k	k	(Lowercase k )
108	6C	01101100	l	l	(Lowercase l )
109	6D	01101101	m	m	(Lowercase m )
110	6E	01101110	n	n	(Lowercase n )
111	6F	01101111	o	o	(Lowercase o )
112	70	01110000	p	p	(Lowercase p )
113	71	01110001	q	q	(Lowercase q )
114	72	01110010	r	r	(Lowercase r )
115	73	01110011	s	s	(Lowercase s )
116	74	01110100	t	t	(Lowercase t )
117	75	01110101	u	u	(Lowercase u )
118	76	01110110	v	v	(Lowercase v )

Dec	Hex	Binary	Symb	Char	Description
119	77	01110111	w	w	(Lowercase w )
120	78	01111000	x	x	(Lowercase x )
121	79	01111001	y	y	(Lowercase y )
122	7A	01111010	z	z	(Lowercase z )
123	7B	01111011	{	{	(curly brackets or braces) (vertical-bar, vbar, vertical line or vertical slash)
124	7C	01111100			(curly brackets or braces)
125	7D	01111101	}	}	(Tilde ; swung dash)
126	7E	01111110	~	~	(Delete)
127	7F	01111111	☐	DEL	(Delete)
128	80	10000000	€	Ç	(Majuscule C-cedilla) (letter "u" with umlaut or diaeresis ; "u- umlaut")
129	81	10000001	•	ü	(letter "e" with acute accent or "e- acute")
130	82	10000010	,	é	(letter "a" with circumflex accent or "a- circumflex")
131	83	10000011	f	â	(letter "a" with umlaut or diaeresis ; "a- umlaut")
132	84	10000100	„	ä	(letter "a" with grave accent)
133	85	10000101	...	à	(letter "a" with a ring)
134	86	10000110	†	å	(Minuscule c-cedilla)
135	87	10000111	‡	ç	(letter "e" with circumflex accent or "e- circumflex")
136	88	10001000	^	ê	(letter "e" with umlaut or diaeresis ; "e- umlaut")
137	89	10001001	‰	ë	(letter "e" with grave accent)
138	8A	10001010	Š	è	(letter "i" with umlaut or diaeresis ; "i- umlaut")
139	8B	10001011	‹	ï	(letter "i" with circumflex accent or "i- circumflex")
140	8C	10001100	Œ	î	(letter "i" with grave accent)
141	8D	10001101	•	ì	(letter "A" with umlaut or diaeresis ; "A-umlaut")
142	8E	10001110	Ž	Ä	(Capital letter "A" with a ring)
143	8F	10001111	•	Å	(Capital letter "E" with acute accent or "E-acute")
144	90	10010000	•	É	(Latin diphthong "ae" in lowercase)
145	91	10010001	‘	æ	(Latin diphthong "AE" in uppercase)
146	92	10010010	’	Æ	(letter "o" with circumflex accent or "o- circumflex")
147	93	10010011	“	ô	(letter "o" with umlaut or diaeresis ; "o- umlaut")
148	94	10010100	”	ö	(letter "o" with grave accent)
149	95	10010101	•	ò	(letter "u" with circumflex accent or "u- circumflex")
150	96	10010110	—	û	(letter "u" with grave accent)
151	97	10010111	—	ù	(letter "u" with grave accent)



Dec	Hex	Binary	Symb	Char	Description
152	98	10011000	~	ÿ	(Lowercase letter "y" with diaeresis)
153	99	10011001	™	Ö	(letter "O" with umlaut or diaeresis ; "O-umlaut")
154	9A	10011010	š	Ü	(letter "U" with umlaut or diaeresis ; "U-umlaut")
155	9B	10011011	›	ø	(slashed zero or empty set)
156	9C	10011100	œ	£	(Pound sign ; symbol for the pound sterling)
157	9D	10011101	•	Ø	(slashed zero or empty set)
158	9E	10011110	ž	×	(multiplication sign)
159	9F	10011111	Ÿ	f	(function sign ; f with hook sign ; florin sign )
160	A0	10100000		á	(letter "a" with acute accent or "a- acute")
161	A1	10100001	ı	í	(letter "i" with acute accent or "i- acute")
162	A2	10100010	ç	ó	(letter "o" with acute accent or "o- acute")
163	A3	10100011	£	ú	(letter "u" with acute accent or "u- acute")
164	A4	10100100	¤	ñ	(letter "n" with tilde ; enye)
165	A5	10100101	¥	Ñ	(letter "N" with tilde ; enye)
166	A6	10100110	ı	ª	(feminine ordinal indicator)
167	A7	10100111	§	º	(masculine ordinal indicator)
168	A8	10101000	¨	¿	(Inverted question marks)
169	A9	10101001	©	®	(Registered trademark symbol)
170	AA	10101010	ª	¬	(Logical negation symbol)
171	AB	10101011	«	½	(One half)
172	AC	10101100	¬	¼	(Quarter or one fourth)
173	AD	10101101		¡	(Inverted exclamation marks)
174	AE	10101110	®	«	(Angle quotes or guillemets)
175	AF	10101111	¬	»	(Guillemets or angle quotes)
176	B0	10110000	º	░	
177	B1	10110001	±	▒	
178	B2	10110010	²	▓	
179	B3	10110011	³	┃	(Box drawing character)
180	B4	10110100	´	┃	(Box drawing character)
181	B5	10110101	μ	Á	(Capital letter "A" with acute accent or "A-acute")
182	B6	10110110	¶	Â	(letter "A" with circumflex accent or "A-circumflex")
183	B7	10110111	·	À	(letter "A" with grave accent)
184	B8	10111000	„	©	(Copyright symbol)
185	B9	10111001	¡	┃	(Box drawing character)
186	BA	10111010	º	┃	(Box drawing character)

Dec	Hex	Binary	Symb	Char	Description
187	BB	10111011	»	☐	(Box drawing character)
188	BC	10111100	¼	☐	(Box drawing character)
189	BD	10111101	½	¢	(Cent symbol)
190	BE	10111110	¾	¥	(YEN and YUAN sign)
191	BF	10111111	¿	☐	(Box drawing character)
192	C0	11000000	À	☐	(Box drawing character)
193	C1	11000001	Á	☐	(Box drawing character)
194	C2	11000010	Â	☐	(Box drawing character)
195	C3	11000011	Ã	☐	(Box drawing character)
196	C4	11000100	Ä	—	(Box drawing character)
197	C5	11000101	Å	+	(Box drawing character)
198	C6	11000110	Æ	ã	(Lowercase letter "a" with tilde or "a-tilde") (Capital letter "A" with tilde or "A-tilde")
199	C7	11000111	Ç	Ã	(Box drawing character)
200	C8	11001000	È	☐	(Box drawing character)
201	C9	11001001	É	☐	(Box drawing character)
202	CA	11001010	Ê	☐	(Box drawing character)
203	CB	11001011	Ë	☐	(Box drawing character)
204	CC	11001100	Ì	☐	(Box drawing character)
205	CD	11001101	Í	=	(Box drawing character)
206	CE	11001110	Î	☐	(Box drawing character)
207	CF	11001111	Ï	¤	(generic currency sign)
208	D0	11010000	Ð	ð	(Lowercase letter "eth")
209	D1	11010001	Ñ	Ð	(Capital letter "Eth")
210	D2	11010010	Ò	Ê	(letter "E" with circumflex accent or "E-circumflex") (letter "E" with umlaut or diaeresis ; "E-umlaut")
211	D3	11010011	Ó	Ë	(letter "E" with grave accent)
212	D4	11010100	Ô	È	(lowercase dot less i)
213	D5	11010101	Õ	ı	(Capital letter "I" with acute accent or "I-acute") (letter "I" with circumflex accent or "I-circumflex") (letter "I" with umlaut or diaeresis ; "I-umlaut")
214	D6	11010110	Ö	Í	(letter "I" with umlaut or diaeresis ; "I-umlaut")
215	D7	11010111	×	Î	(letter "I" with grave accent)
216	D8	11011000	Ø	Ï	(Box drawing character)
217	D9	11011001	Ù	☐	(Box drawing character)
218	DA	11011010	Ú	☐	(Box drawing character)
219	DB	11011011	Û	■	(Block)
220	DC	11011100	Ü	■	(Bottom half block)
221	DD	11011101	Ý	⋮	(vertical broken bar)
222	DE	11011110	Þ	Ì	(letter "I" with grave accent)
223	DF	11011111	ß	■	(Top half block)

Dec	Hex	Binary	Symb	Char	Description
224	E0	11100000	à	Ó	(Capital letter "O" with acute accent or "O-acute")
225	E1	11100001	á	ß	(letter "Eszett" ; "scharfes S" or "sharp S")
226	E2	11100010	â	Ô	(letter "O" with circumflex accent or "O-circumflex")
227	E3	11100011	ã	Ò	(letter "O" with grave accent)
228	E4	11100100	ä	õ	(letter "o" with tilde or "o-tilde")
229	E5	11100101	å	Õ	(letter "O" with tilde or "O-tilde")
230	E6	11100110	æ	μ	(Lowercase letter "Mu" ; micro sign or micron)
231	E7	11100111	ç	þ	(Lowercase letter "Thorn")
232	E8	11101000	è	Ɔ	(Capital letter "thorn")
233	E9	11101001	é	Ú	(Capital letter "U" with acute accent or "U-acute")
234	EA	11101010	ê	Û	(letter "U" with circumflex accent or "U-circumflex")
235	EB	11101011	ë	Ü	(letter "U" with grave accent)
236	EC	11101100	ì	ý	(Lowercase letter "y" with acute accent)
237	ED	11101101	í	Ý	(Capital letter "Y" with acute accent)
238	EE	11101110	î	-	(macron symbol)
239	EF	11101111	ï	˘	(Acute accent)
240	F0	11110000	ð		(Hyphen)
241	F1	11110001	ñ	±	(Plus-minus sign)
242	F2	11110010	ò	=	(underline or underscore)
243	F3	11110011	ó	¾	(three quarters)
244	F4	11110100	ô	¶	(paragraph sign or pilcrow)
245	F5	11110101	õ	§	(Section sign)
246	F6	11110110	ö	÷	(The division sign ; Obelus)
247	F7	11110111	÷	¸	(cedilla)
248	F8	11111000	ø	°	(degree symbol )
249	F9	11111001	ù	¨	(Diaeresis)
250	FA	11111010	ú	·	(Interpunct or space dot)
251	FB	11111011	û	¹	(superscript one)
252	FC	11111100	ü	³	(cube or superscript three)
253	FD	11111101	ý	²	(Square or superscript two)
254	FE	11111110	þ	■	(black square)
255	FF	11111111	ÿ	nbsp	(non-breaking space or no-break space)

#### 2.2.4 Enkripsi One Time Pad

*One Time Pad* adalah salah satu metode kriptografi dengan algoritma simetris. Metode ini awalnya dikembangkan pada tahun 1917 oleh *Major Yoseph*

*Mouborgne* dan *Gilbert Vernam* pada perang dunia ke dua. One Time Pad telah diklaim sebagai satu-satunya algoritma kriptografi sempurna yang tidak dapat dipecahkan [10]. Suatu algoritma dikatakan aman, apabila tidak ada cara untuk menemukan plaintext-nya. Sampai saat ini, hanya algoritma One Time Pad (OTP) yang dinyatakan tidak dapat dipecahkan meskipun diberikan sumber daya yang tidak terbatas.

Dalam skema ini, kunci yang digunakan terdiri dari bilangan acak, yang dimanfaatkan untuk mengenkripsi keseluruhan pesan. Selain itu, kunci yang digunakan untuk enkripsi-dekripsi adalah kunci sekali pakai. Dengan demikian, tidak ada pola tetap yang dapat dianalisis oleh para penyadap. Skema semacam ini tidak dapat dipecahkan karena menghasilkan output yang tidak memiliki hubungan statistik dengan teks biasa. *One Time Pad* adalah utilitas terbatas dan berguna untuk aplikasi yang membutuhkan tingkat keamanan yang sangat tinggi [11]. Secara umum algoritma OTP dapat menggunakan XOR atau modulo (mod) [12], adapun notasi dengan modulo 26 [13] adalah sebagai berikut:

$$C_i = (P_i + K_i) \bmod 26 \quad (2.1)$$

dengan ketentuan:

$C_i$  = cipherteks (teks terenkripsi);

$P_i$  = plaintext;

$K_i$  = kunci random;

26 = jumlah karakter yang dimungkinkan.

Dalam steganografi, keamanan pesan menjadi prioritas, oleh karena itu, untuk meningkatkan keamanan pesan rahasia, konsep *one time pad* digunakan dalam model steganografi teks ini.

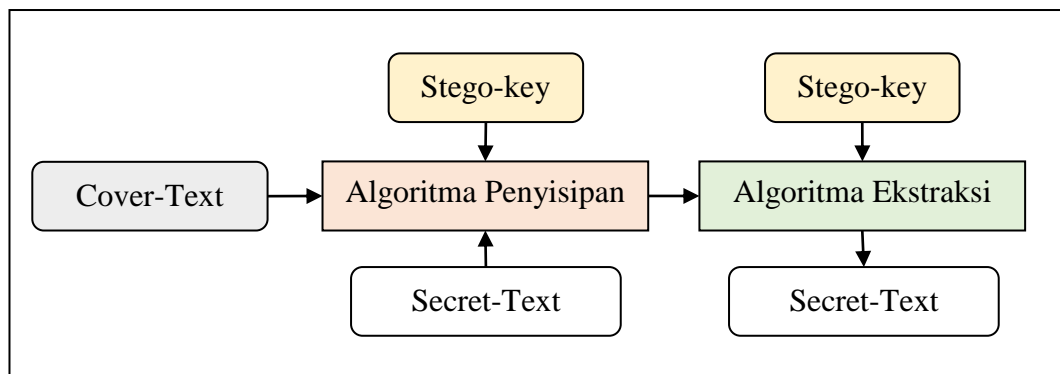
## 2.2.5 Steganografi

### 2.2.5.1 Definisi

Asal muasal kata Steganografi (*steganography*) berasal dari bahasa Yunani yang secara harfiah terdiri dari kata *steganos* berarti tersembunyi dan *graphien* yang diartikan menulis, sehingga steganografi dapat didefinisikan

sebagai tulisan yang tersembunyi. Dari penjabaran secara harfiah tersebut bisa disimpulkan bahwa, Steganografi merupakan sebuah ilmu yang mempelajari cara memanipulasi suatu pesan rahasia di dalam pesan yang lainnya menggunakan cara-cara tertentu sehingga orang yang tidak berkepentingan tidak akan mengetahui bahwa terdapat sebuah informasi rahasia di dalam pesan yang mereka baca [14].

Steganografi merupakan seni menyembunyikan informasi dengan cara mencegah terdeteksinya pesan yang disembunyikan melalui berbagai media antara lain bisa menggunakan gambar, audio, video dan teks. Adapun proses penyembunyian informasi rahasia dalam teknik steganografi diawali dengan mengidentifikasi bit-bit *redundant* dari *cover* mediumnya (yang dapat dimodifikasi tanpa merusak integritas *medium* yang bersangkutan). Proses penyisipan menghasilkan suatu medium stego melalui penggantian bit *redundant* dengan data pesan rahasia yang disembunyikan tersebut [15]. Secara umum proses metode steganografi dapat dijelaskan pada Gambar 2.1.



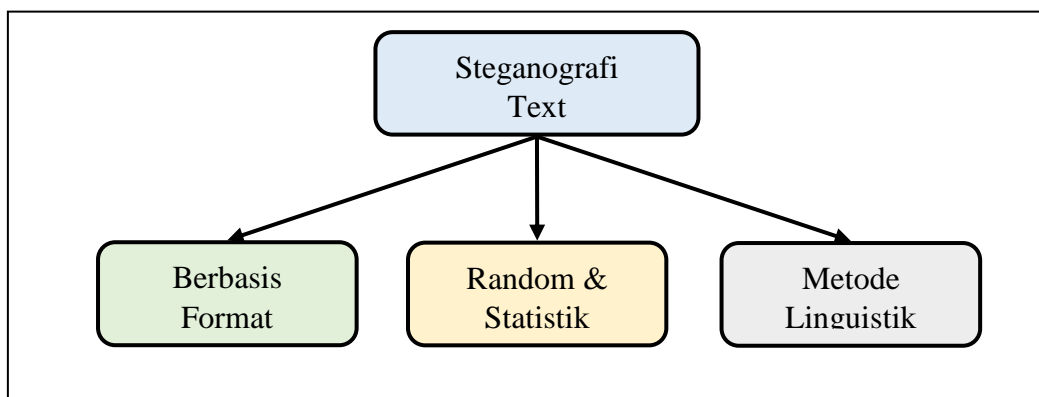
Gambar 2.1 Proses steganografi [15].

Penerapan steganografi teks dapat dilakukan dengan berbagai teknik rekayasa teks mulai dari mengubah format teks yang ada, mengubah kata dalam teks, menggunakan urutan huruf atau karakter secara acak ataupun memanfaatkan perubahan konteks dalam tata bahasa [16].

Metode steganografi teks dipandang paling sulit untuk diterapkan karena sedikitnya redundansi data yang dapat digunakan untuk menyisipkan pesan rahasia, hal ini berbeda jika dibandingkan steganografi dengan menggunakan media *cover* image, audio ataupun video yang mempunyai redundansi yang cukup tinggi.

Sebuah struktur dokumen teks identik dengan apa yang terlihat oleh mata telanjang, sedangkan pada jenis dokumen lainnya seperti image, audio, dan video struktur dokumennya akan berbeda dengan yang terlihat. Oleh karena itu, dalam dokumen tersebut mudah untuk menyembunyikan informasi dengan memanfaatkan perubahan struktur dokumen tanpa membuat perubahan yang nampak pada output steganografi [17]. Perubahan-perubahan data dapat dengan mudah dilakukan tanpa menarik perhatian secara kasat mata pada file image, audio atau audio, namun berbeda halnya dalam file teks, karena perubahan sedikit saja pada tanda baca atau hurufnya dari file teks asli maka dengan mudah dapat ditandai oleh mata telanjang [18]

Steganografi teks secara umum diklasifikasikan ke dalam tiga kategori; *steganografi teks berbasis format*, *steganografi teks Random & statistik*, dan *steganografi teks dengan metode linguistik*.



Gambar 2.2 Metode Steganografi Text [15].

#### A. Steganografi Teks Berbasis Format

*Steganografi teks berbasis format* melibatkan perubahan secara fisik bentuk format teks untuk digunakan dalam penyembunyian informasi. Metode ini memiliki kekurangan tertentu, jika *teks stego* dibuka dengan pengolah kata maka kesalahan pengejaan dan adanya spasi putih tambahan akan terdeteksi. Ukuran huruf yang berubah bisa menimbulkan kecurigaan pada pembaca manusia. Selain itu, jika *plain text* asli tersedia, membandingkan *plain text* ini dengan *teks stego* yang dicurigai akan membuat bagian teks yang dimanipulasi menjadi terlihat [16].

## B. Steganografi Teks Random dan Statistik

*Metode Random dan statistik* adalah metode yang menggunakan metode acak untuk menghasilkan *teks sampel* secara otomatis sesuai dengan statistik. Metode ini menggunakan tata bahasa yang dijadikan sebagai sebuah media untuk menghasilkan teks sampel dalam bahasa alami tertentu.

## C. Steganografi Teks Linguistik

*Metode Linguistik* adalah metode yang menggunakan sifat bahasa tersebut untuk memodifikasi text. Metode Linguistik menggunakan struktur bahasa sebagai tempat untuk menyembunyikan informasi atau text rahasia.

### 2.2.5.2 Fungsi Steganografi

Steganografi banyak digunakan dan diimplementasikan dalam berbagai bidang antara lain [4]:

- Penyempurnaan Struktur Data

Adakalanya suatu hal yang mengharuskan penambahan sebuah informasi ke dalam suatu struktur data, dimana struktur yang ada tidak mendukung format penambahan yang baru sehingga mengharuskan pembaruan *software*, tentunya hal ini cukup merepotkan namun dengan steganografi hal ini menjadi mungkin untuk dilakukan misal penyematan informasi ke dalam sebuah foto.

- Watermark yang Handal

Penggunaan *watermark* pada sebuah media dimaksudkan untuk mengukuhkan keasliannya sehingga dengan sedemikian rupa penanaman watermark dibuat sekuat mungkin, namun adakalanya juga watermark dibuat dengan maksud jika ada sedikit saja perubahan pada media tersebut maka keasliannya akan menjadi hilang, steganografi mampu untuk memenuhi kebutuhan model watermark tersebut.

- Alat Pelacakan Dokumen

Steganografi dapat digunakan sebagai pengidentifikasi file. Jika suatu dokumen bocor atau terdistribusi kepada orang yang tidak berwenang, dokumen itu bisa dilacak kembali ke pemilik yang sah. Dengan menambahkan tag individu secara tersembunyi ke setiap dokumen.

- Otentikasi File

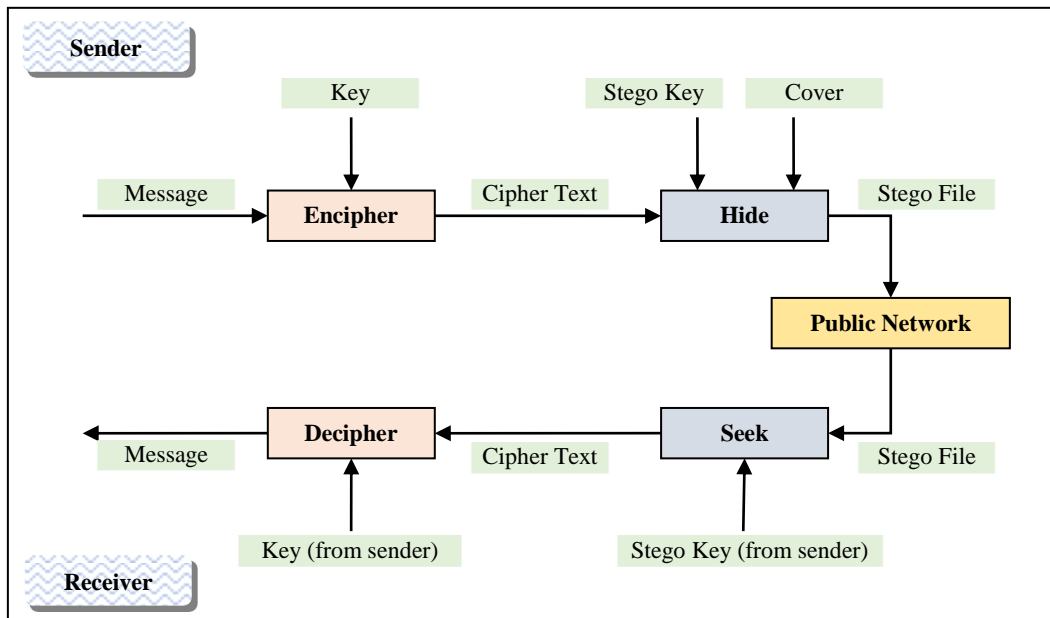
Informasi tersembunyi yang digabungkan dengan sebuah file bisa berfungsi sebagai tandatangan digital yang berfungsi menyertifikasi keotentikannya.

- Komunikasi Pribadi

Dengan kemampuannya menyembunyikan pesan dengan sendirinya steganografi bermanfaat untuk digunakan sebagai alat komunikasi secara pribadi.

### 2.2.5.3 Metode Penyembunyian Data dalam Paragraf

Metode ini terdiri dari empat bagian: (1) Fungsi enkripsi, yakni mengenkripsi pesan yang akan disembunyikan menggunakan skema *One Time Pad*, (2) Fungsi penyembunyian, yaitu menyembunyikan pesan acak menggunakan *kunci stego*, (3) Fungsi Pencarian, yaitu mengekstrak informasi tersembunyi dari *teks stego* menggunakan *kunci stego*, dan (4) Fungsi Dekripsi, yaitu mendekripsi pesan yang diekstrak menggunakan kunci rahasia [5].



Gambar 2.3 Metode Steganografi Teks dengan Penyembunyian Paragraf [5]

Pendekatan metode ini memanfaatkan *teks sampul* yang bisa diambil dari sumber manapun (misalnya paragraf dari surat kabar / buku) yang telah



ditentukan sebelumnya. Metode ini bekerja dengan menyembunyikan pesan menggunakan huruf awal dan akhir dari kata-kata dari sebuah *teks sampul*. Pendekatan ini bekerja pada *digit biner* sebuah karakter ASCII. Setelah mengubah teks sandi (*ciphertext*) menjadi *Bit-stream*, masing-masing bit disembunyikan dengan mengambil satu kata dari *teks sampul* dengan menggunakan huruf awal atau akhir dari sebuah kata hal ini tergantung pada bit yang akan disembunyikan. Bit 0 atau 1 disembunyikan dengan cara membaca sebuah kata secara berurutan dari *teks sampul* yang didalamnya termasuk huruf awal atau huruf akhir dari kata *kunci stego*. Sebuah kata yang memiliki huruf awal dan akhir yang sama maka akan dilewati (tidak digunakan), karena tidak ada perubahan yang terjadi pada sampulnya.

#### *Algoritma Penyembunyian (Enkripsi)*

1. Dapatkan file sampul;
2. Mengkonversi file input ke angka biner (bin);
3. Membaca bit (x) dari bin;
4. Membaca sebuah kata dari file sampul dan menuliskan di file stego;
5. Jika huruf awal dan akhir dari sebuah kata sama, maka baca kata berikutnya dari file cover dan tulis di file stego;
6. s = start huruf dari kata dan e = end dari kata;
7. If x = 0, tuliskan s pada stego key;
8. Else if x = 1, tulis e di stego key;
9. Ulangi langkah 3 sampai 8 sampai akhir dari file bin;
10. Kirim file stego dan stego key ke receiver.

#### *Algoritma Pencarian (Dekripsi)*

1. Baca karakter (c) dari stego key;
2. Baca sebuah kata dari file stego;
3. Jika huruf start dan end dari kata sama, lompat kata itu dan baca kata selanjutnya dari file stego;
4. Dapatkan huruf awal (s) dan huruf akhir (e) dari kata;
5. If c = s, then bit b = 0;

6. Else if  $c = e$ , then bit  $b = 1$ ;
7. Tuliskan  $b$  dalam sebuah file;
8. Jalankan langkah di atas berulang kali sampai akhir dari stego key;
9. Mengkonversi file menjadi karakter yang ekuivalen.

Contoh:

Sebuah pesan rahasia "try" untuk disembunyikan dalam file sampul. Setelah mengenkripsi pesan, teks sandi yang dihasilkan adalah "wµc". Gambar 2.4 menunjukkan file cover dan Gambar 2.5 menunjukkan file stego. Kita dapat melihat bahwa kedua file itu persis sama.

The house itself looked empty. The doors and windows were locked. The front verandah bare. Unfurnished. But the skyblue Plymouth with chrome tailfins was still parked outside, and inside, Baby Kochamma was still alive. She was Rahel's baby grandaunt, her grandfather's younger sister. Her name was really Navomi, Navomi Ipe, but everybody called her Baby. She became Baby Kochamma when she was old enough to be an aunt.

Gambar 2.4 Cover File[5]

The house itself looked empty. The doors and windows were locked. The front verandah bare. Unfurnished. But the skyblue Plymouth with chrome tailfins was still parked outside, and inside, Baby Kochamma was still alive. She was Rahel's baby grandaunt, her grandfather's younger sister. Her name was really Navomi, Navomi Ipe, but everybody called her Baby. She became Baby Kochamma when she was old enough to be an aunt.

Gambar 2.5 Stego File [5]

#### 2.2.5.4 Pemetaan Binary Digit (Bit)

Pengertian pemetaan bit (*bit mapping*) adalah memetakan nilai *digit biner* (bit) pada setiap karakter ASCII sebagai metode penyematan steganografi

teks untuk menghasilkan urutan nomor yang digunakan sebagai *kunci stego*. Pemetaan ini menggunakan 7 bit nilai setiap karakter baik sebagai *teks rahasia* maupun sebagai *teks sampul*, contoh *digit biner* huruf Z adalah 1011010 digunakan sebagai *teks rahasia*, kemudian ABCDEFG sebagai *teks sampul* dengan digit binernya 1000001, 1000010, 1000011, 1000100, 1000101, 1000110, 1000111, maka pemetaannya adalah menyematkan setiap *digit biner* teks rahasia berurutan dari kiri ke kanan kedalam salah satu *digit biner* yang sama di setiap karakter teks sampul dengan mencari secara berurutan mulai dari kanan ke kiri, dan nomor urutan dicatat sebagai *kunci stego*. Langkah-langkah pemetaannya seperti dijelaskan pada gambar berikut

	1	1	1	3	2	2	4	→	Stego-Key
	1	2	3	4	5	6	7		Sequence number
Z	1	0	1	1	0	1	0	→	Teks rahasia
	7	6	5	4	3	2	1		Sequence number
A	1	0	0	0	0	0	1	}	Teks sampul
B	1	0	0	0	0	1	0		
C	1	0	0	0	0	1	1		
D	1	0	0	0	1	0	0		
E	1	0	0	0	1	0	1		
F	1	0	0	0	1	1	0		
G	1	0	0	0	1	1	1		
ASCII	Binary digits (bit) value								

Gambar 2.6 Metode pemetaan bit

### 2.2.6 Kapasitas

Untuk mengukur daya tampung teks sampul dalam menyembunyikan teks rahasia maka perlu diukur rasio kapasitasnya, dimana Kapasitas didefinisikan sebagai kemampuan teks sampul dapat menyembunyikan teks rahasia. Rasio kapasitas dihitung dengan membagi jumlah bit teks rahasia terhadap teks sampul

dalam bit [19]. Rasio kapasitas dapat diketahui dengan menggunakan persamaan berikut:

$$\text{Rasio Kapasitas} = \frac{\text{bit secret text}}{\text{bit cover text}} \times 100 \% \quad (2.2)$$

### 2.2.7 Jaro-Winkler Distance

*Jaro-Winkler Distance* merupakan bagian dari *Jaro Distance Metrics* yakni suatu algoritma *Similarity Distance* untuk mengukur kesamaan antara dua string, biasanya algoritma ini digunakan untuk mendeteksi duplikasi suatu file. Hasil perhitungan *Jaro-Winkler Distance* menghasilkan rentang skor 0 sampai dengan 1, skor 0 menunjukkan tidak adanya kesamaan antara dua string yang dibandingkan, sedangkan skor 1 menunjukkan bahwa dua string yang dibandingkan mempunyai isi yang sama persis.

Dasar perhitungan dari algoritma ini memiliki tiga bagian yaitu: menghitung panjang string, menemukan jumlah karakter yang sama di dalam dua string, dan menemukan jumlah transposisi [20]. Misal untuk menghitung jarak ( $d_j$ ) antara dua string yaitu string  $s_1$  dan string  $s_2$  [21] maka digunakan persamaan berikut:

$$d_j = \frac{1}{3} \times \left( \frac{m}{|s_1|} + \frac{m}{|s_2|} + \frac{m-t}{m} \right) \quad (2.2)$$

di mana  $|s_1|$  adalah panjang string ke-1 (teks sampul),  $|s_2|$  adalah panjang string ke-2 (teks stego),  $m$  adalah jumlah karakter yang sama pada  $|s_1|$  dan  $|s_2|$  dan  $t$  adalah jumlah transposisi. Masing-masing karakter pada string  $|s_1|$  dan string  $|s_2|$  dinyatakan cocok/mirip jika kedua string tersebut sama dan tidak melebihi ambang batas dari

$$\left\lfloor \frac{\max |s_1|, |s_2|}{2} \right\rfloor - 1 \quad (2.3)$$

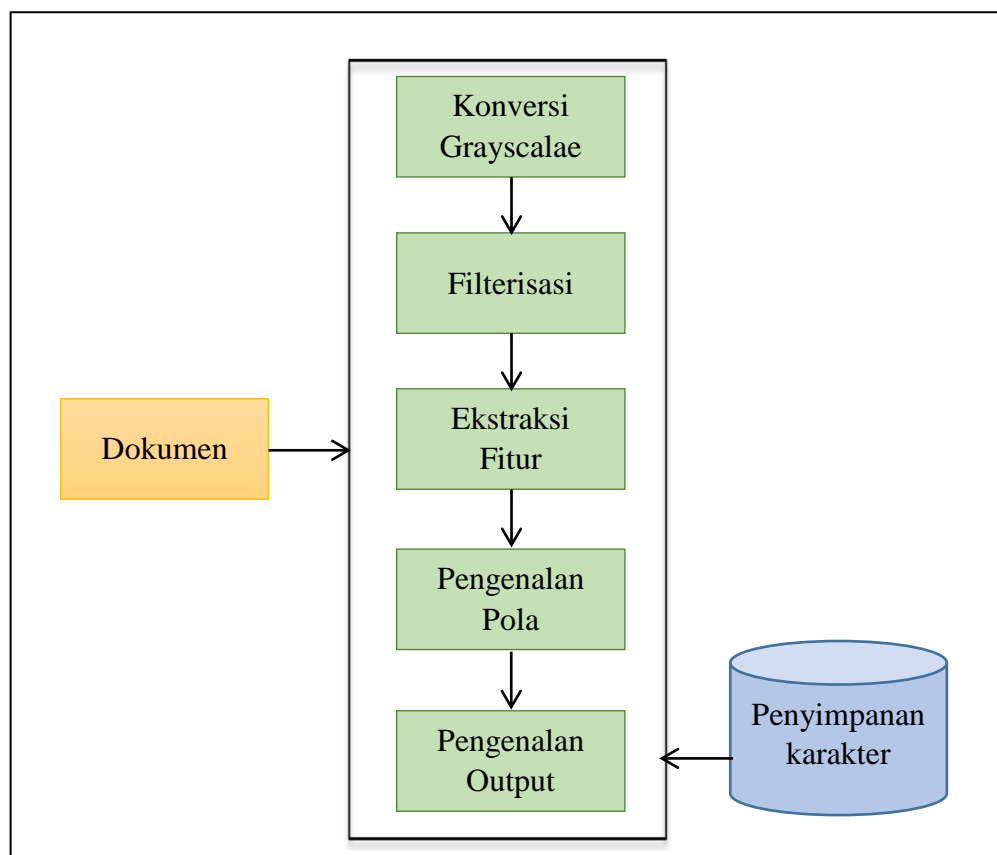
Jaro-Winkler Distance ( $d_w$ ) menggunakan skala prefix ( $p$ ) yang menyediakan awalan pada serangkaian string, dengan persamaan berikut,

$$d_w = d_j + (lp(1 - d_j)) \quad (2.4)$$

di mana  $d_j$  adalah hasil dari perhitungan kesamaan string  $s_1$  dan  $s_2$ , dan 1 adalah panjang prefix secara umum pada permulaan string hingga maksimum empat karakter,  $p$  adalah faktor skala konstan, nilai default untuk konstanta sesuai dengan algoritma Winkler adalah  $p = 0,1$ . Sedangkan proses perhitungan menghasilkan rentang nilai dari 0 hingga 1, dimana nilai 0 menunjukkan tidak ada kemiripan antara dua string sedangkan nilai 1 menyatakan dua string yang dibandingkan adalah sama persis.

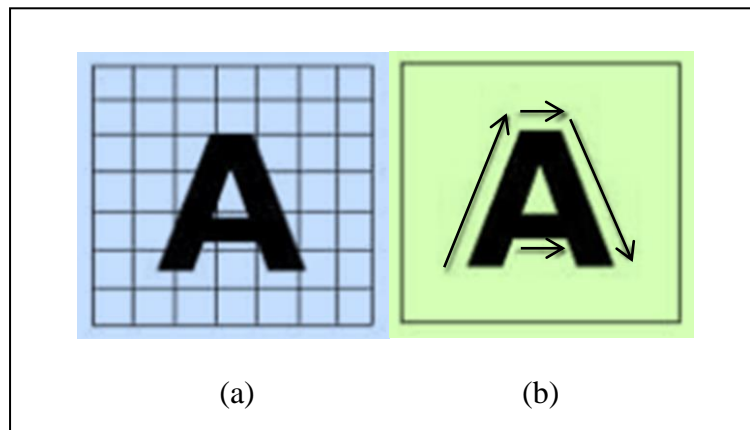
### 2.2.8 Optical Character Recognition (OCR)

*Optical character recognition* (OCR) merupakan sebuah proses pengenalan karakter pada dokumen cetak maupun dokumen hasil pemindaian untuk dikonversi menjadi karakter ASCII [22]. OCR bekerja dengan cara mendeteksi sebuah objek berupa tulisan untuk dapat dikenali setiap karakternya sehingga dapat dibaca oleh komputer. Proses OCR dapat dicermati pada diagram blok yang ditunjukkan dengan Gambar 2.7.



Gambar 2.7 Diagram blok OCR [22]

Implementasi metode OCR dapat dilakukan secara *offline* dan juga secara *online*, OCR secara offline diawali dengan cara menghasilkan dokumen dalam bentuk digital melalui proses pemindaian untuk disimpan dalam sistem komputer dan selanjutnya dilakukan proses pengenalan komputer, sedangkan OCR secara online pengenalan karakter dilakukan secara bersamaan dengan proses pemindaian [23]. Pada Gambar 2.8 dapat dilihat perbedaan proses OCR secara offline dan secara online.



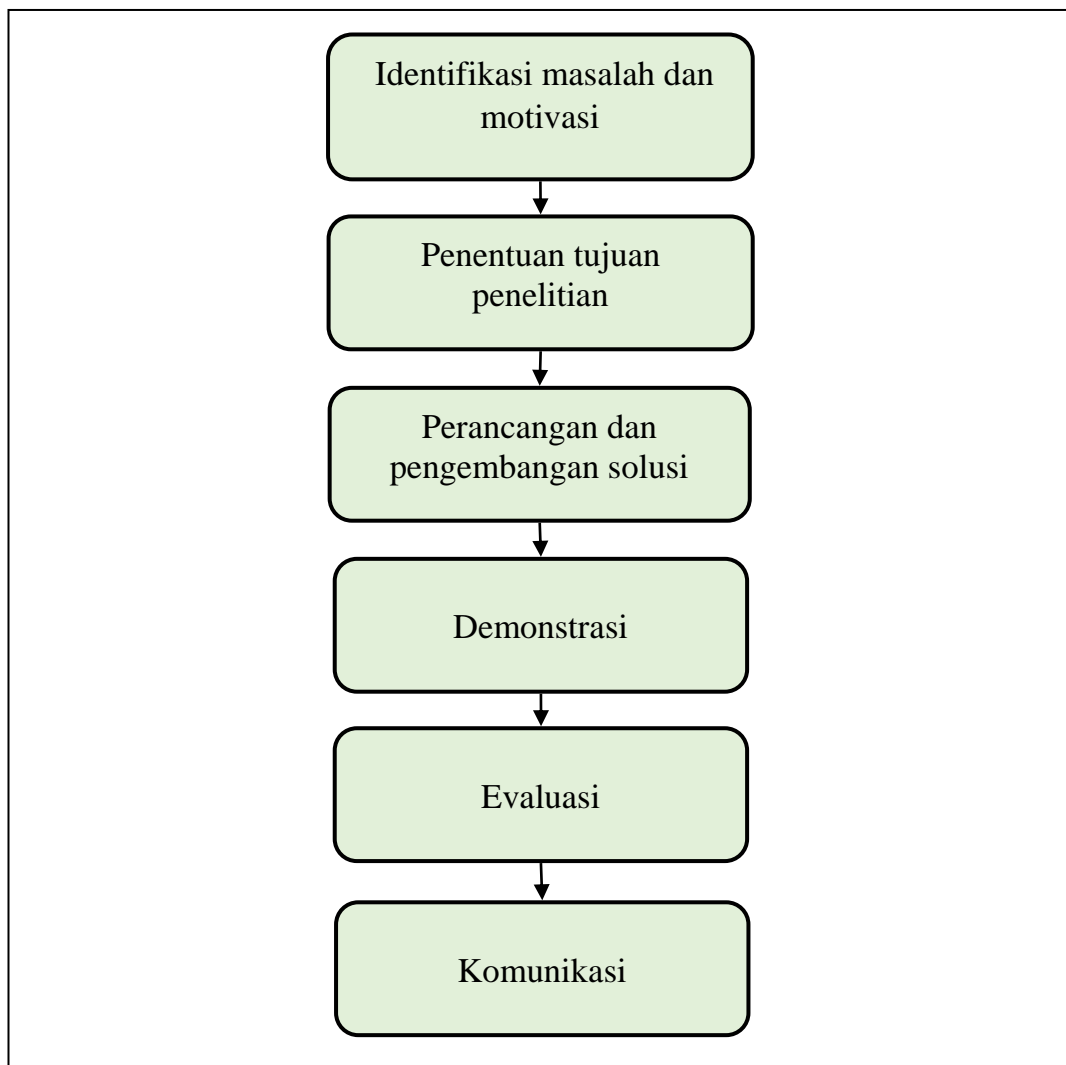
Gambar 2.8 (a) Pengenalan karakter secara offline, dan (b) secara online [23]

## BAB 3

### METODOLOGI PENELITIAN

#### 3.1 Skema Penelitian

Penelitian ini dilakukan dengan menggunakan pendekatan *Design Science Research Methodology* (DSRM) [24] yang terdiri dari enam proses tahapan seperti ditunjukkan pada Gambar 3.1 yaitu: mengidentifikasi masalah dan motivasi, penentuan tujuan penelitian, perancangan dan pengembangan solusi, demonstrasi, evaluasi dan komunikasi.



Gambar 3.1 Skema Penelitian dengan metode DSRM [24]

### 3.1.1 Identifikasi Masalah dan Motivasi

Pada tahap awal ini dilakukan pengidentifikasian permasalahan dengan studi literatur merujuk pada berbagai buku dan makalah dari jurnal internasional maupun makalah seminar internasional yang membahas tentang steganografi teks.

Dari studi literatur diketahui bahwa keamanan plain text dapat dilakukan dengan menerapkan teknik steganografi teks dengan metode *Hiding Data in Paragraphs* [5], metode ini memungkinkan penyembunyian sebuah pesan rahasia kedalam format teks biasa (*plain text*) yakni dengan merekayasa digit biner setiap karakter ASCII (*American Standard Code for Information Interchange*). Dari hipotesa awal penggunaan steganografi teks dengan metode tersebut bisa digunakan sebagai cara untuk mengetahui jika sebuah informasi yang berbasis teks telah mengalami perubahan atau tidak, dan penelitian menggunakan metode tersebut secara spesifik dengan tujuan keamanan *plain text* belum ditemukan.

### 3.1.2 Penentuan Tujuan Penelitian

Berdasarkan identifikasi permasalahan ditetapkan tujuan penelitian yaitu mengembangkan teknik steganografi teks dengan metode *Hiding Data in Paragraphs* [5] dengan cara menyisipkan *teks rahasia* melalui pemetaan bit pada karakter ASCII, dengan proses awal mengenkripsi *teks rahasia* terlebih dahulu dengan metode *One Time Pad* (OTP) dan menghasilkan *ciphertext*, kemudian *teks rahasia* yang telah menjadi *ciphertext* beserta *teks sampul* sebagai media penyisipannya dikonversi kedalam nilai kode binernya.

Dengan hipotesa awal bahwa plain text yang telah disisipi *teks rahasia* dapat berfungsi sebagai media penyampai pesan rahasia yang dapat diekstrak kembali, dan *teks sampul* yang digunakan dapat berfungsi sebagai *integrity text* yang keotentikasiannya dapat dikonfirmasi.

### 3.1.3 Perancangan dan Pengembangan Solusi

Dengan ditetapkannya tujuan penelitian maka disusun perancangan sebuah sistem dengan metode Pemetaan Bit pada karakter ASCII (*Bit Mapping*), konsepsi metode penelitian diimplementasikan dengan antar muka berbasis web menggunakan pemrograman *Hypertext Preprocessor* (PHP).



### 3.1.4 Demonstrasi

Berdasarkan rancangan yang telah disusun perlu dilakukan demonstrasi untuk melihat kesesuaian rancangan dengan tujuan dari penelitian yaitu dengan simulasi berbagai uji coba metode berdasarkan konsep.

### 3.1.5 Evaluasi

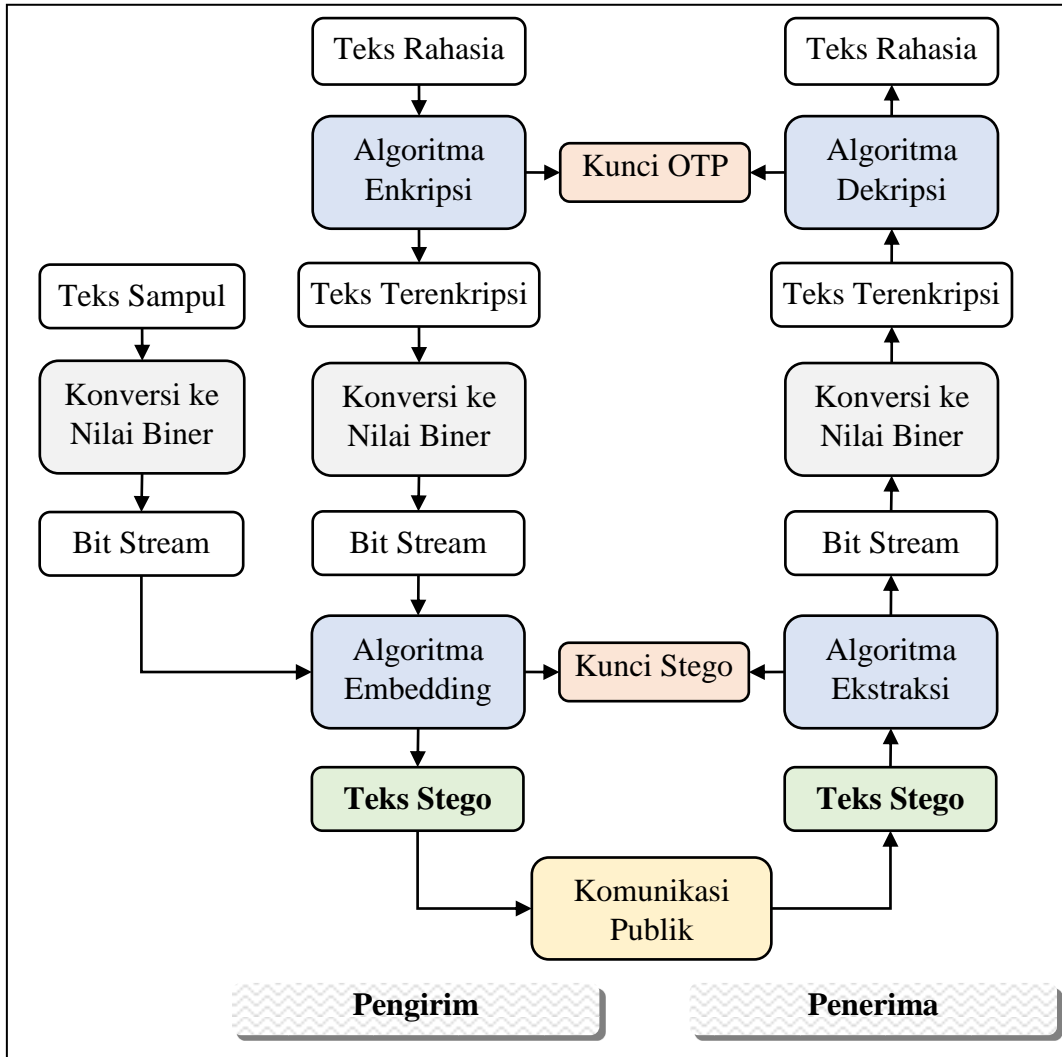
Pada tahap evaluasi dilakukan pengujian terhadap hasil metode *Pemetaan Bit pada karakter ASCII* dengan cara proses ekstraksi kembali *teks rahasia* yang telah disematkan kedalam *teks sampul*, kemudian dilakukan pengukuran daya tampung *teks sampul* terhadap *teks rahasia* dengan penghitungan  $\text{Kapasitas} = \text{teks sampul} / \text{teks rahasia}$ . Selanjutnya juga diukur tingkat kesamaan antara *teks sampul* dengan *teks stego* yang dihasilkan menggunakan teknik *Similarity Distance* dengan metode *Jaro-Winkler Distance*.

### 3.1.6 Komunikasi

Penyampaian hasil penelitian diajukan sebagai laporan tesis dalam bentuk buku tesis pada Institut Teknologi Sepuluh Nopember Surabaya. Laporan penelitian tersebut diharapkan dapat memberikan kontribusi dalam peningkatan keamanan dalam bidang informasi berbasis plain text dan dapat menjadi rujukan penelitian dan pengembangan selanjutnya.

## 3.2 Analisis Sistem

Dalam penerapan steganografi teks dengan metode *Pemetaan Bit*, media yang digunakan sebagai teks sampul adalah teks dalam format *plain text* yang bisa menggunakan seluruh karakter ASCII, sedangkan untuk teks rahasia menggunakan abjad A sampai dengan Z, kemudian baik teks sampul maupun teks rahasia dikonversi menjadi angka biner, dimana sebelumnya dilakukan enkripsi pada teks rahasia menggunakan *One Time Pad* (OTP) modulo 26, yang selanjutnya dilakukan proses penyisipan menghasilkan teks stego dan juga kunci stego yang terdiri dari angka 1,2,3,4,5,6,7. Skema teks steganografi dengan metode penyisipan pemetaan bit dapat dilihat pada Gambar 3.2.



Gambar 3.2 Skema steganografi teks dengan metode pemetaan bit

Secara umum proses steganografi teks dengan metode pemetaan bit dapat diuraikan sebagai berikut; misal himpunan huruf-huruf abjad  $A = \{A, B, C, \dots, Y, Z\}$ , himpunan bilangan bulat modulo 26 yaitu  $Z_{26}$ , teks rahasia  $M = a_1, a_2, \dots, a_m$  dengan  $a_i \in A$ , Kunci OTP  $K = k_1, k_2, \dots, k_m$  dengan  $k_i \in A$ , kemudian fungsi  $Asc(x)$  adalah fungsi untuk mengubah  $x \in A$  menjadi 7 digit terakhir kode biner ASCII, dan  $f$  adalah fungsi bijektif  $f: A \rightarrow Z_{26}$  yang mengubah huruf abjad menjadi representasi integernya. Teks terenkripsi  $E$  dapat diperoleh dengan

$$E = f^{-1}([f(M) + f(K)] \bmod 26) \quad (3.1)$$

selanjutnya representasi biner dari  $E$  dapat diperoleh dengan

$$E_b = Asc(e), \forall e \in E \quad (3.2)$$

dengan  $|E_b| = 7m$ . Misal digunakan teks sampul  $C = c_1, c_2, \dots, c_n$ , dengan  $n \geq 7m$ . Bentuk biner dari  $C$  dapat diperoleh dengan

$$C_b = Asc(c), \forall c \in C \quad (3.3)$$

Selanjutnya dilakukan penyisipan dari digit-digit biner  $E_b$  ke digit-digit biner  $C_b$  dengan aturan  $E_b$  dibaca dari kiri ke kanan,  $C_b$  dibaca dari karakter pertama (pada 7 digit pertama) dari kanan ke kiri. Misal  $e_i \in E_b$  dan  $c_j \in C_b$ , dengan  $i = 1, 2, \dots, 7m$ .  $j = 1, 2, \dots, 7n$ . Selanjutnya diperoleh kunci stego dari proses penyisipan, yaitu  $W = w_1, w_2, \dots, w_{7n}$ , dengan aturan  $w_k = j$  jika  $e_i = c_j$  dan juga didapatkan sebuah teks stego  $S$ .

Proses ekstraksi dari teks stego adalah sebagai berikut; misal penerima pesan telah menerima teks stego  $S$ , kunci stego  $W$  dan kunci OTP  $K$ . Maka langkah pertama; teks stego  $S$  diubah ke bentuk biner  $S_b$  dengan

$$S_b = Asc(s), \forall s \in S \quad (3.4)$$

pesan rahasia  $E_b$  dapat diekstrak dari  $S_b$  menggunakan kunci stego  $W$ . Misal  $w_k \in W$ , lalu dari setiap karakter (7 digit biner) di  $S_b$ , diekstrak menjadi  $E_b$  dengan mengambil digit indeks ke  $w_k$  pada  $S_b$ , dengan  $k = 1, 2, \dots, 7n$ . Selanjutnya  $E_b$  diubah menjadi  $E$  dengan

$$E = Asc^{-1}(e), \forall e \in E_b \quad (3.5)$$

kemudian  $E$  diubah menjadi  $M$  dengan

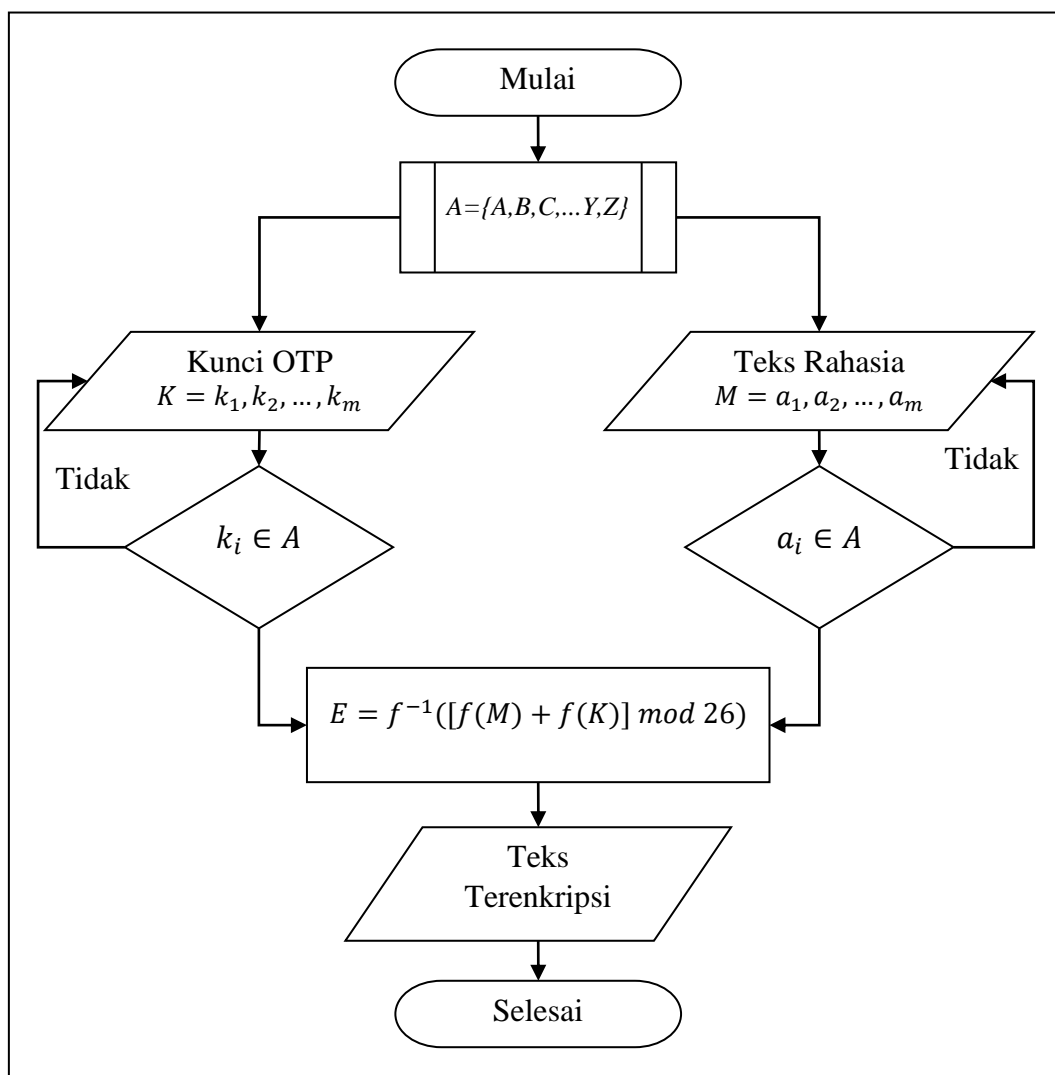
$$M = f^{-1}([f(E) - f(K)] \bmod 26) \quad (3.6)$$

maka diperoleh kembali teks rahasia yang sebelumnya disisipkan pada teks sampul.

### 3.3 Perancangan Sistem

#### 3.3.1 Enkripsi dengan One Time Pad (OTP) menggunakan Modulo 26

Proses awal yang dilakukan adalah mengenkripsi teks rahasia dengan tujuan memberikan pengamanan yang berlapis, jika ada pihak luar yang berhasil mengetahui adanya pesan rahasia pada teks sampul yang mengakibatkan teks rahasia diketahui maka pihak luar tersebut masih perlu mendekripsi teks rahasia untuk mengetahui isi pesan rahasia tersebut. Sistem enkripsi dengan OTP mod 26 dapat dilihat pada Gambar 3.3.



Gambar 3.3 Diagram alir enkripsi menggunakan OTP mod 26

Dari Gambar 3.3 dapat dijelaskan proses enkripsi sebagai berikut:

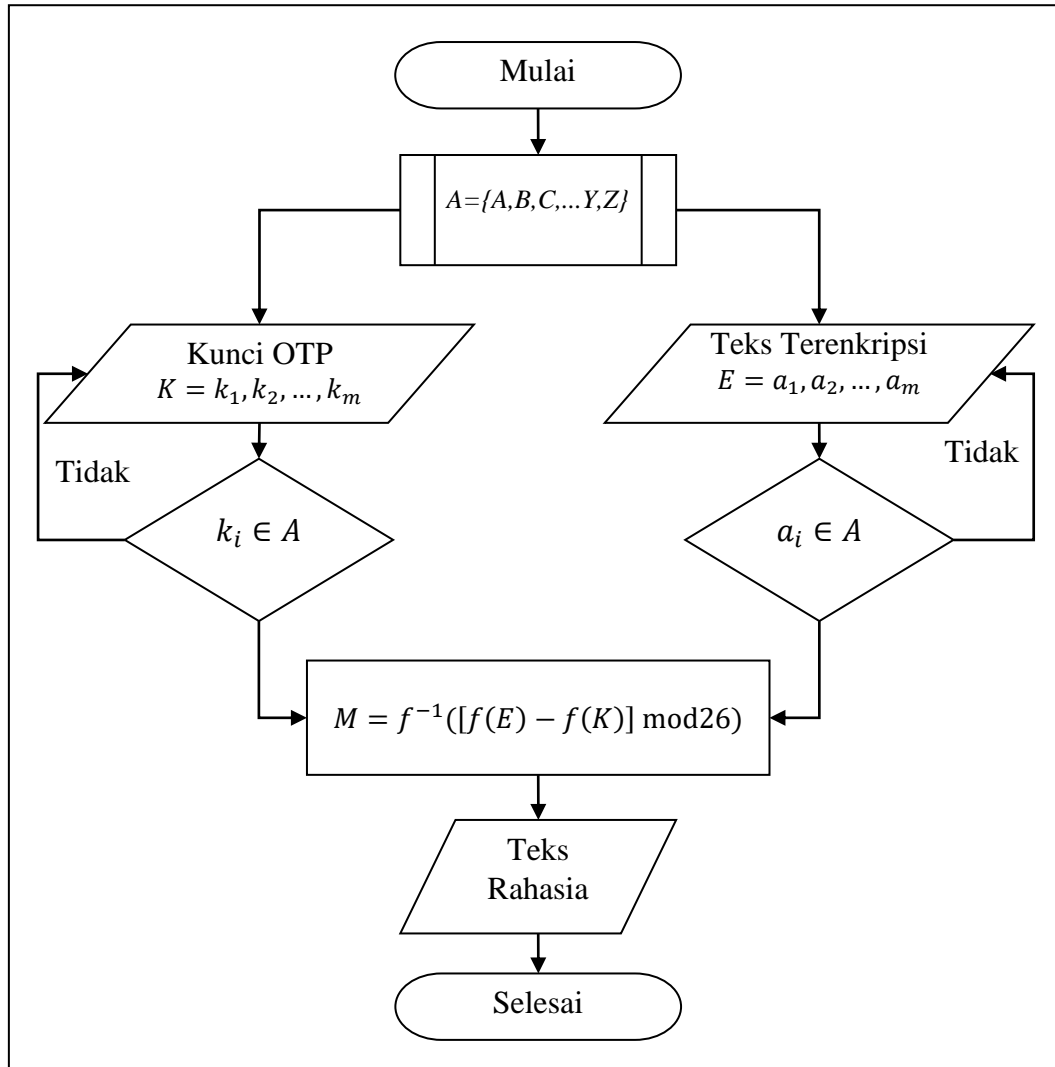
1. Sebuah plain text terdiri dari himpunan huruf-huruf abjad  $A = \{A, B, C, \dots, Y, Z\}$ , dimana himpunan yang berjumlah 26 huruf ini adalah himpunan bilangan bulat modulo 26 yakni  $Z_{26}$ ;
2. Teks rahasia  $M = a_1, a_2, \dots, a_m$  dengan ketentuan  $a_i \in A$ ;
3. Kunci OTP  $K = k_1, k_2, \dots, k_m$  dengan ketentuan  $k_i \in A$ ;
4. Selanjutnya  $f$  adalah fungsi bijektif  $f: A \rightarrow Z_{26}$  yang mengubah huruf abjad menjadi representasi integernya;
5. Untuk mengenkripsi teks rahasia dengan kunci OTP digunakan persamaan  $E = f^{-1}([f(M) + f(K)] \bmod 26)$ ;
6. Didapatkan sebuah teks terenkripsi yang mempunyai jumlah karakter yang sama sebagaimana teks rahasia.

### 3.3.2 Dekripsi dengan One Time Pad (OTP) menggunakan Modulo 26

Untuk mendapatkan kembali bentuk teks rahasia dari teks terenkripsi maka perlu dilakukan proses dekripsi dari teks terenkripsi tersebut, proses dekripsi dapat dicermati pada Gambar 3.4. yang memuat diagram alir pemrosesan dekripsi dengan penjelasan sebagai berikut:

1. Didapatkan Sebuah teks terenkripsi yang terdiri dari himpunan huruf-huruf abjad  $A = \{A, B, C, \dots, Y, Z\}$ , dimana himpunan ini adalah himpunan bilangan bulat modulo 26 yaitu  $Z_{26}$ ;
2. Teks terenkripsi  $E = a_1, a_2, \dots, a_m$  dengan ketentuan  $a_i \in A$ ;
3. Kunci OTP  $K = k_1, k_2, \dots, k_m$  dengan ketentuan  $k_i \in A$ ;
4. Selanjutnya  $f$  adalah fungsi bijektif  $f: A \rightarrow Z_{26}$  yang mengubah huruf abjad menjadi representasi integernya;
5. Untuk mendekripsi teks terenkripsi maka digunakan kunci OTP yang telah ditentukan sebelumnya dan dilakukan perhitungan menggunakan persamaan  $M = f^{-1}([f(E) - f(K)] \bmod 26)$ ;

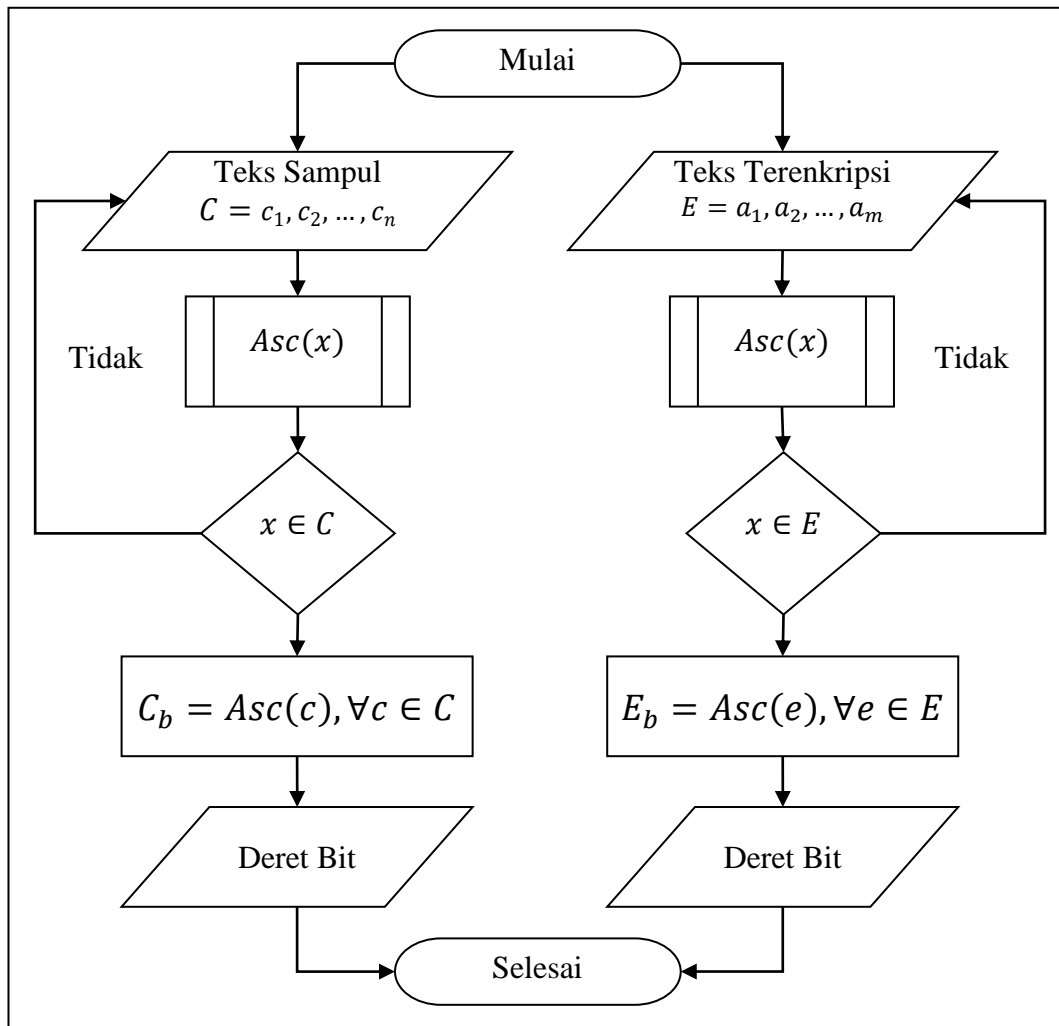
6. Didapatkan sebuah teks rahasia yang sebelumnya dalam bentuk teks terenkripsi.



Gambar 3.4 Diagram alir dekripsi menggunakan OTP mod 26

### 3.3.3 Konversi ASCII ke Nilai Biner Digit (Bit)

Metode pemetaan bit bekerja pada nilai *bianary digit* (bit) baik pada teks rahasia maupun pada teks sampul sehingga perlu dilakukan konversi dari karakter ASCII ke dalam nilai binernya. Proses konversi ditunjukkan dengan diagram alir pada Gambar 3.5.



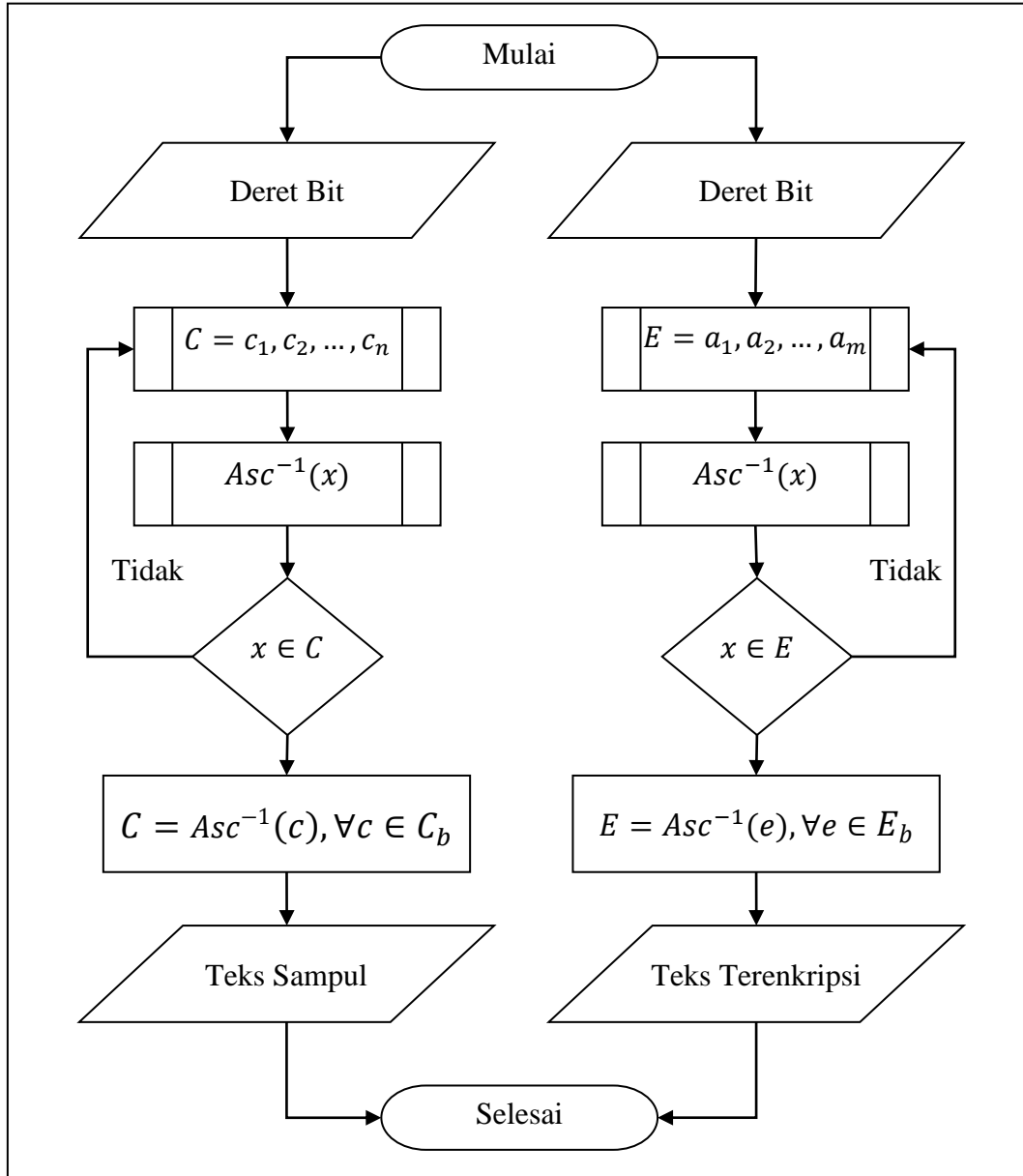
Gambar 3.5 Diagram alir konversi karakter ASCII ke dalam nilai binernya

Dari diagram alir pada Gambar 3.5 dapat dilihat ada dua proses konversi data yang berbeda secara bersamaan yaitu teks sampel dan teks terenkripsi namun dengan metode yang sama. Penjelasan langkah-langkah konversi dari diagram alir tersebut diambil salah satu yaitu konversi teks sampel ke dalam nilai binernya:

1. Didapatkan sebuah teks sampel  $C = c_1, c_2, \dots, c_n$ ;
2. Digunakan fungsi  $Asc(x)$  yang berguna mengubah setiap karakter teks sampel menjadi 7 digit terakhir nilai binernya dengan ketentuan  $x \in C$ ;
3. Konversi teks sampel ke dalam nilai binernya menggunakan persamaan  $C_b = Asc(c), \forall c \in C$
4. Dari proses komputasi tersebut dihasilkan deret digit biner (bit) yang merupakan representasi karakter ASCII dari teks sampel.

### 3.3.4 Konversi Nilai Biner Digit (Bit) ke ASCII

Untuk mengembalikan lagi karakter ASCII dari nilai binernya maka dilakukan konversi kembali dari deret bit menjadi plain text seperti ditunjukkan pada Gambar 3.6.



Gambar 3.6 Diagram alir konversi deret bit kedalam karakter ASCII

Dari diagram alir pada Gambar 3.6 juga terlihat ada dua proses konversi data yang berbeda secara bersamaan yaitu deret bit teks sampul dan deret bit teks terenkripsi namun dengan metode yang sama. Penjelasan langkah-langkah



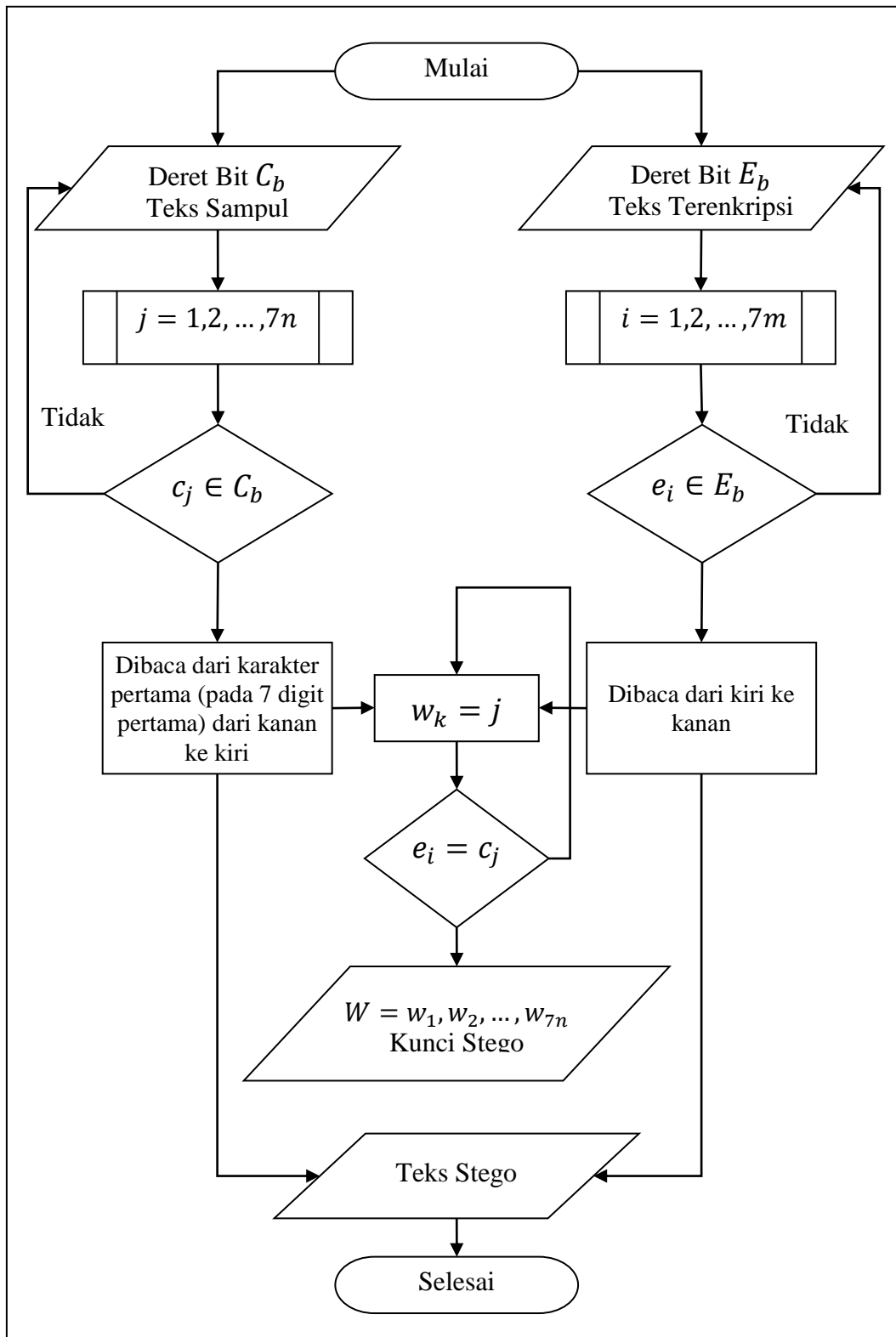
konversi dari diagram alir tersebut diambil salah satu yaitu konversi deret bit teks sampel ke dalam representasi karakter ASCII:

1. Didapatkan deret bit teks sampel  $C = c_1, c_2, \dots, c_n$ ;
2. Digunakan fungsi  $Asc^{-1}(x)$  yang berfungsi mengubah setiap 7 bit dari deret bit teks sampel dengan ketentuan  $x \in C$ ;
3. Konversi deret bit teks sampel kedalam representasi karakter ASCII menggunakan persamaan  $C = Asc^{-1}(c), \forall c \in C_b$
4. Dari proses komputasi tersebut dihasilkan karakter ASCII teks sampel yang dibentuk dari deret biner digit (bit) yang merupakan representasi teks sampel.

### 3.3.5 Metode Pemetaan Binary Digit (Bit)

Pemetaan *binary digit* (bit) adalah memetakan bit pada setiap karakter ASCII sebagai metode steganografi teks untuk menghasilkan teks stego dan kunci stego. Gambar 3.7 menunjukkan diagram alir proses pemetaan bit dengan penjelasan sebagai berikut:

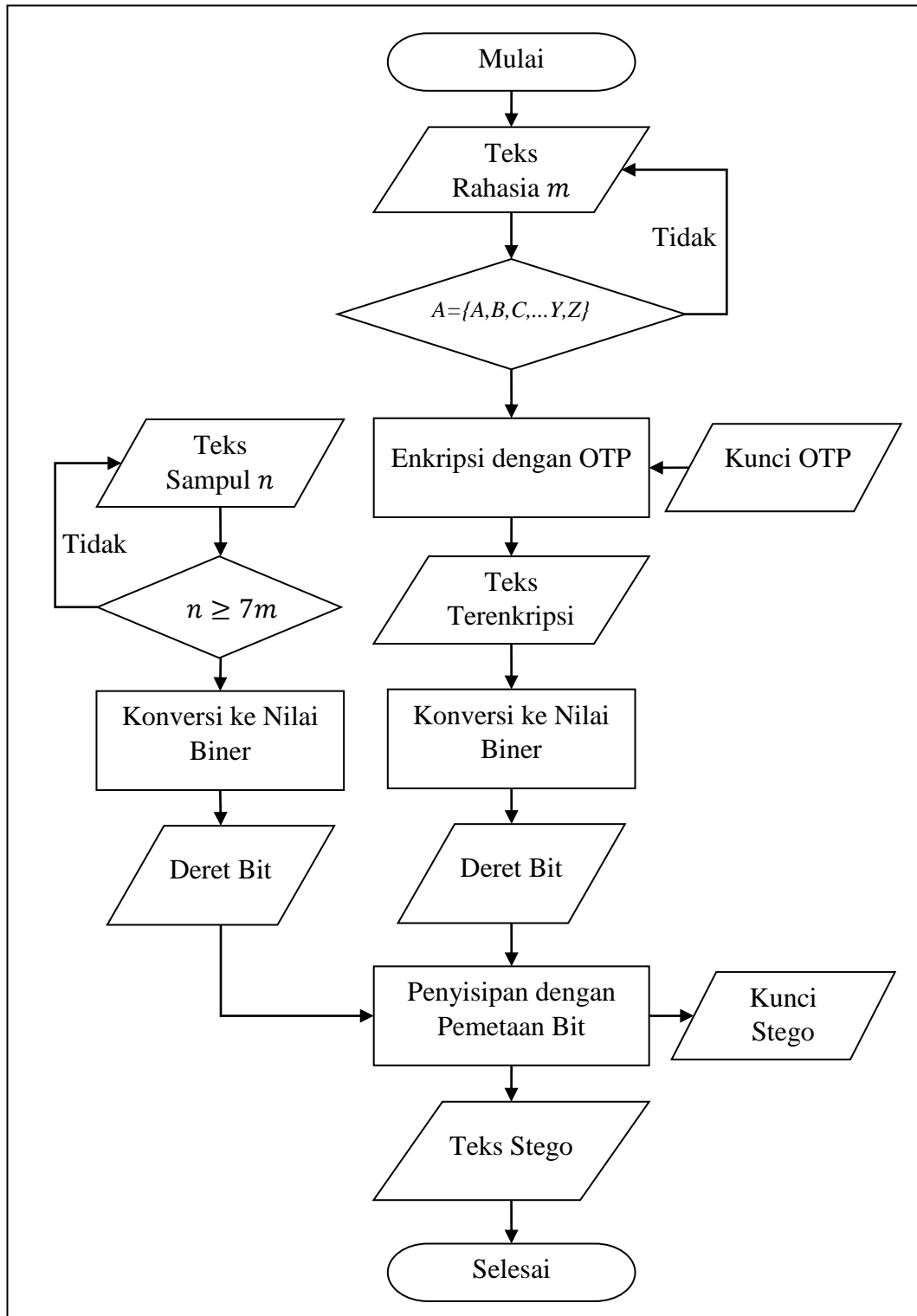
1. Didapatkan deret bit teks sampel  $C_b$  dimana  $c_j \in C_b$  dengan  $j = 1, 2, \dots, 7n$ ;
2. Didapatkan deret bit teks terenkripsi  $E_b$  dimana  $e_i \in E_b$  dengan  $i = 1, 2, \dots, 7m$ ;
3. Selanjutnya dilakukan penyisipan  $E_b$  terhadap  $C_b$ , dengan mengambil satu persatu seluruh  $E_b$  dengan aturan  $E_b$  dibaca dari kiri ke kanan;
4. Kemudian  $E_b$  yang disisipkan ke  $C_b$ , dengan ketentuan  $C_b$  dibaca dari karakter pertama (pada 7 digit pertama) dan dibaca dari kanan ke kiri;
5. Dari proses komputasi tersebut didapatkan kunci stego  $W = w_1, w_2, \dots, w_{7n}$  dengan ketentuan  $w_k = j$  jika  $e_i = c_j$ ;
6. Dan juga sebagai hasil akhir didapatkan teks stego.



Gambar 3.7 Diagram alir Pemetaan *Binary Digit* (Bit)

### 3.3.6 Proses Penyisipan

Proses penyisipan teks rahasia kedalam teks sampul dengan metode pemetaan bit dapat dilihat pada diagram alir di Gambar 3.8.



Gambar 3.8 Diagram alir penyisipan teks rahasia dengan pemetaan bit

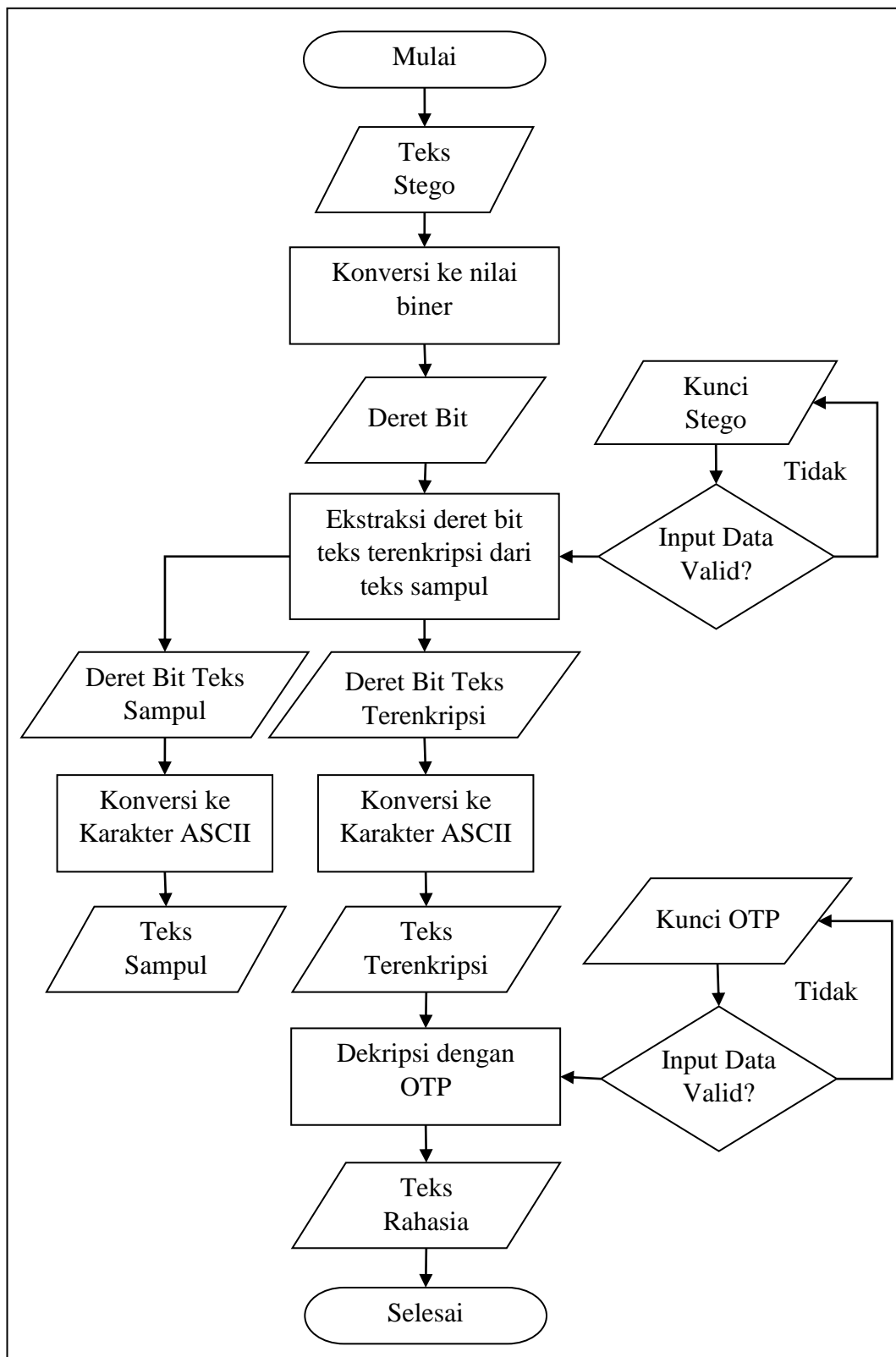
Dari Gambar 3.8 dengan diagram alir penyisipan teks steganografi dengan pemetaan bit dapat dijelaskan sebagai berikut:

1. Didapatkan teks rahasia  $m$  yang merupakan himpunan  $A = \{A, B, C, \dots, Y, Z\}$ ;
2. Kemudian dilakukan enkripsi dengan OTP mod 26 dan menghasilkan teks terenkripsi;
3. Selanjutnya teks terenkripsi dikonversi menjadi deret bit;
4. Didapatkan teks sampul  $n$  yang terdiri dari karakter ASCII dengan ketentuan  $n \geq 7m$ ;
5. Kemudian teks sampul  $n$  dikonversi menjadi deret bit;
6. Selanjutnya dilakukan penyisipan dari deret bit teks terenkripsi ke dalam deret bit teks sampul dengan pemetaan bit;
7. Dari proses komputasi didapatkan teks stego beserta kunci stego.

### 3.3.7 Proses Ekstraksi

Ekstraksi teks rahasia dari teks sampul menggunakan metode pemetaan bit sebagaimana ditunjukkan pada diagram alir di Gambar 3.9. Proses komputasi ekstraksi dijelaskan sebagaimana berikut:

1. Didapatkan teks stego kemudian dikonversi ke nilai binernya dan didapatkan deret bit;
2. Didapatkan kunci stego yang selanjutnya digunakan untuk mengekstrak deret bit teks terenkripsi dari deret bit teks sampul;
3. Didapatkan deret bit teks terenkripsi dan deret bit teks sampul;
4. Masing-masing deret bit dikonversi ke karakter ASCII menjadi teks terenkripsi dan teks sampul;
5. Teks terenkripsi didekripsi menggunakan OTP mod 26 dan didapatkan teks rahasia.



Gambar 3.9 Diagram alir ekstraksi teks rahasia dengan pemetaan bit

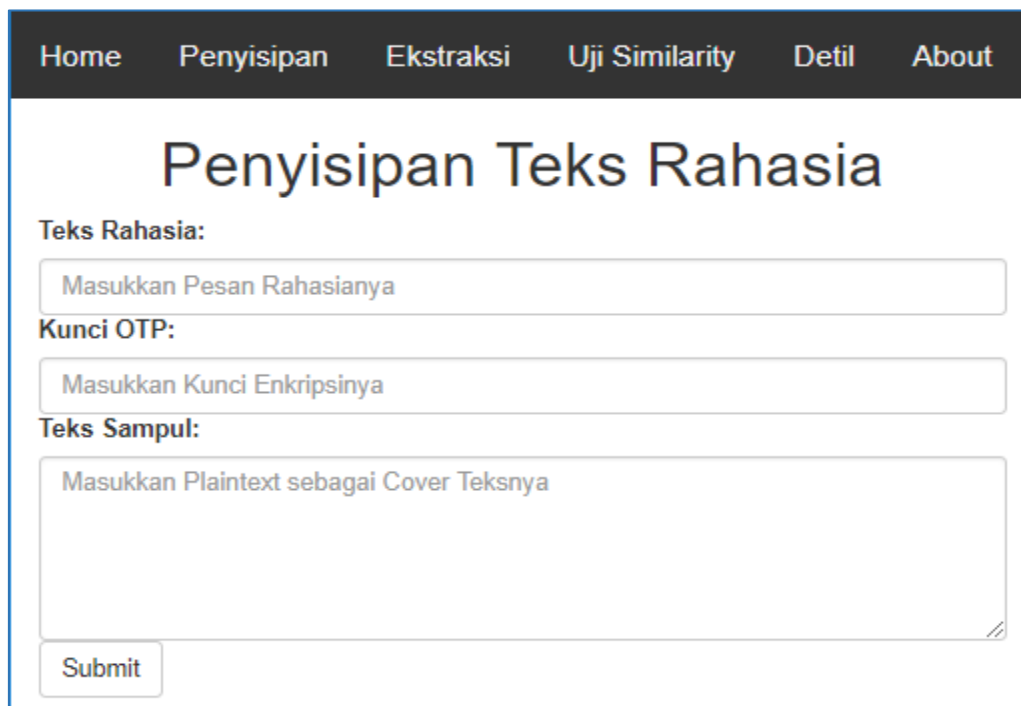
### 3.4 Implementasi Sistem

Dari pemodelan sistem yang telah dirancang sebelumnya dilakukan implementasi teks steganografi dengan metode pemetaan bit menggunakan pemrograman *Hypertext Preprocessor* (PHP) dengan beberapa bagian pemrosesan.

#### 3.4.1 Penyisipan Teks Rahasia

Bagian antar muka *Penyisipan Teks Rahasia* merupakan bagian pemrosesan penyisipan teks rahasia kedalam teks sampul dengan kunci OTP dengan inputan yang dapat dilihat pada Gambar 3.10, terdiri dari tiga inputan dan satu tombol eksekusi yaitu:

1. Teks Rahasia, inputan yang digunakan untuk memasukkan teks rahasia;
2. Kunci OTP, digunakan untuk memasukkan teks yang digunakan sebagai kunci OTP mod 26;
3. Teks Sampul, berfungsi untuk memasukkan plain text yang digunakan sebagai teks sampul;
4. Submit, tombol yang digunakan untuk memulai proses penyisipan teks setelah semua inputan telah lengkap dimasukkan.



Gambar 3.10 Antar muka Penyisipan Teks Rahasia

### 3.4.2 Ekstraksi Teks Rahasia

Bagian antar muka *Ekstraksi Teks Rahasia* merupakan bagian untuk proses ekstraksi teks sampul untuk mendapatkan teks rahasia dari teks sampulnya dengan menggunakan kunci stego dan kunci OTP. Detil bagian dapat dilihat pada Gambar 3.11, terdiri dari tiga inputan dan satu tombol eksekusi yaitu:

1. Teks Stego, yang digunakan untuk memasukkan teks stego yang ditentukan;
2. Kunci Stego, digunakan untuk memasukkan kunci stego yang sebelumnya didapat dari proses peyisipan teks rahasia;
3. Kunci OTP, digunakan untuk memasukkan kunci OTP yang juga telah ditentukan sebelumnya.
4. Submit, tombol yang digunakan untuk memulai proses ekstraksi teks rahasia setelah semua inputan telah lengkap dimasukkan.

Home   Penyisipan   Ekstraksi   Uji Similarity   Detil   About

## Ekstraksi Teks Rahasia

**Teks Stego:**

Masukkan Steganografi Teksnya

**Kunci Stego:**

Masukkan Kunci Steganografinya

**Kunci OTP:**

Masukkan Kunci OTP

Submit

Gambar 3.11 Antar muka Ekstraksi Teks Rahasia

### 3.4.3 Detil Proses Pemetaan Bit

Pada bagian antar muka *Detil Proses Pemetaan Bit* mempunyai tampilan yang hampir sama dengan antar muka Penyisipan Teks Rahasia namun dari proses yang dilakukan menghasilkan informasi detil tentang proses penyisipan teks rahasia kedalam teks sampul. Inputan antar muka ini dapat dilihat pada Gambar 3.12, terdiri dari tiga inputan dan satu tombol eksekusi yaitu:

1. Teks Rahasia, yang digunakan untuk memasukkan teks rahasia;
2. Kunci OTP, digunakan untuk memasukkan teks yang digunakan sebagai kunci OTP mod 26;
3. Teks Sampul, berfungsi untuk memasukkan plain text yang digunakan sebagai teks sampul;
4. Submit, tombol yang digunakan untuk memulai proses penyisipan teks setelah semua inputan telah lengkap dimasukkan.



Gambar 3.12 Antar muka Detil Proses Pemetaan Bit

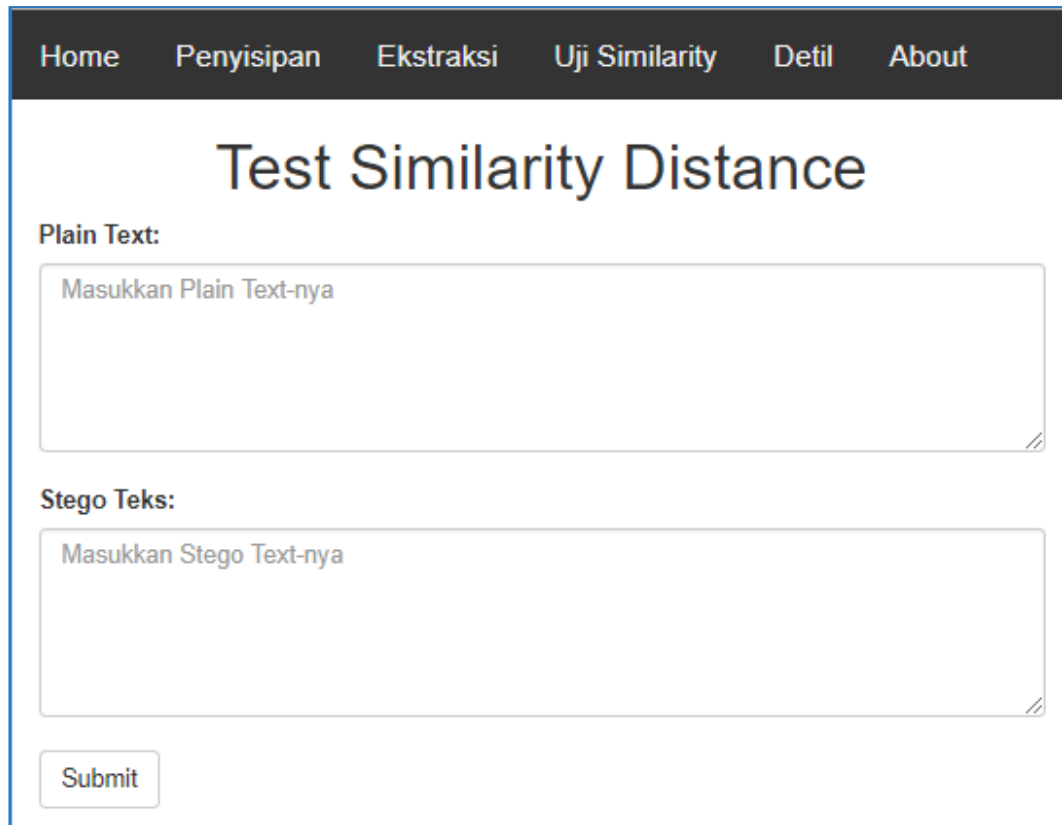
### 3.4.4 Uji Similarity

Pada bagian antar muka *Uji Similarity Proses Pemetaan Bit* berfungsi untuk menguji kemiripan antara *plain text* dan *teks stego*. Inputan antar muka ini



dapat dilihat pada Gambar 3.13, terdiri dari dua inputan dan satu tombol eksekusi yaitu:

1. Plain Text, digunakan untuk memasukkan plain text;
2. Stego Text, digunakan untuk memasukkan teks stego;
3. Submit, tombol yang digunakan untuk memulai proses pengujian setelah semua inputan telah lengkap dimasukkan.



The image shows a web application interface titled "Test Similarity Distance". At the top, there is a dark navigation bar with white text links: "Home", "Penyisipan", "Ekstraksi", "Uji Similarity", "Detil", and "About". Below the navigation bar, the main heading "Test Similarity Distance" is centered in a large, bold, black font. Underneath the heading, there are two text input fields. The first field is labeled "Plain Text:" and contains the placeholder text "Masukkan Plain Text-nya". The second field is labeled "Stego Teks:" and contains the placeholder text "Masukkan Stego Text-nya". Below these two fields, there is a "Submit" button. The entire interface is enclosed in a light gray border.

Gambar 3.13 Antar muka Uji Similarity Pemetaan Bit

*Halaman ini sengaja dikosongkan*

## BAB 4

### HASIL DAN PEMBAHASAN

Pada bab ini dijelaskan dan dibahas hasil proses steganografi teks dengan metode pemetaan *binary digit* (bit) dimana metode diimplementasikan pada *plain text* berbasis karakter ASCII (*American Standard Code for Information Interchange*) dengan diawali proses enkripsi pada teks rahasia yang akan disisipkan kedalam teks sampul, kemudian masing-masing baik teks yang telah terenkripsi sebagai teks rahasia maupun teks sampul dikonversi ke nilai biner menjadi deret biner, selanjutnya dilakukan penyisipan dan menghasilkan teks stego beserta kunci stegonya.

#### 4.1 Penerapan Metode Pemetaan Bit

Hasil penerapan steganografi teks dengan metode pemetaan bit memuat beberapa bagian pembahasan yang mengacu pada metodologi penelitian yang telah dipaparkan sebelumnya.

##### 4.1.1 Enkripsi dan Dekripsi OTP Mod 26

*One Time Pad* (OTP) modulo 26 merupakan metode *enkripsi simetris* yang artinya kunci yang digunakan untuk mengenkripsi adalah kunci yang juga digunakan untuk mendekripsi.

###### 4.1.1.1 Implementasi Enkripsi dan Dekripsi OTP Mod 26

Berikut ini adalah hasil implementasi metode enkripsi dan dekripsi dimana format kunci yang digunakan untuk enkripsi OTP modulo 26 ditunjukkan pada Gambar 4.1, sedangkan persamaan (4.1) menunjukkan formula untuk enkripsi, dan untuk dekripsi dapat dilihat pada persamaan (4.2).

A	B	C	D	E	F	G	H	I	J	K	L	M
0	1	2	3	4	5	6	7	8	9	10	11	12
N	O	P	Q	R	S	T	U	V	W	X	Y	Z
13	14	15	16	17	18	19	20	21	22	23	24	25

Gambar 4.1 Format kunci OTP Modulo 26, dengan huruf kapital dan huruf kecil berlaku sama

$$E = f^{-1}([f(M) + f(K)] \bmod 26) \quad (4.1)$$

$$M = f^{-1}([f(E) - f(K)] \bmod 26) \quad (4.2)$$

Dengan keterangan sebagai berikut:

$E$  = Teks terenkripsi

$M$  = Teks biasa

$K$  = Kunci OTP

$\bmod 26$  = jumlah abjad yang menjadi kunci

$f^{-1}$  = fungsi yang mengubah kedalam nilai integernya

#### 4.1.1.2 Hasil Uji Coba Enkripsi dan Dekripsi OTP Mod 26

Hasil uji coba enkripsi dan dekripsi dengan teks Visi Kementerian Agama Republik Indonesia [25] dengan sejumlah 207 karakter, dengan teks rahasia ditunjukkan pada Gambar 4.2, teks terenkripsi Gambar 4.3, kunci OTP Gambar 4.4 dan plain text hasil dekripsi dapat dilihat pada Gambar 4.5.

VisikementerianAgamaRepublikIndonesiaTerwujudnyaMasyarakatIndonesiaYangTaatBeragamaRukunCerdasdanSejahteraLahirBatinDalamRangkaMewujudkanIndonesiaYangBerdaulatMandiriDanBerkepribadianBerlandaskanGotongRoyong

Gambar 4.2 Teks Rahasia Visi Kementerian Agama RI

IrrnemtehkjainaFbuqhVuudogqpPidsenxzfNvlbaayksGaShmdgafXgaXvjxeirunVbpbOhfzHhwjjkelRjykvSallemuygLxpfnaJujOkguoOxukiYgqhpAsxyvaBukkrqleejCruigclbqNpyqKhygiyoumQtuypmjFgpHtfsmjpcutzywnEwwogskjcvkySbujpdTjhbWj

Gambar 4.3 Teks terenkripsi Visi Kementerian Agama RI

njvfuihaurfjannfvueheqfjnvifhvaerjfrfurufgrehfiaghufgjfnghpigjrezmnxbcvvhfggdfjdkslapoqiQWuieurytttgfghfdjkdzmxnxbcvvgfhdjskslapqoqiWuiewuerutyttqpplkjdhdieutethvhvbcgcpoiuuyuttwqWadsfdghjklklmbvxcvxnjd

Gambar 4.4 Kunci OTP yang dibuat terlebih dahulu secara random

Visi kementerian Agama Republik Indonesia Terwujudnya Masyarakat Indonesia Yang Taat Beragama, Rukun, Cerdas dan Sejahtera Lahir Batin Dalam Rangka Mewujudkan Indonesia Yang Berdaulat, Mandiri, dan Berkepribadian Berlandaskan Gotong Royong

Gambar 4.5 Teks rahasia hasil dekripsi

#### 4.1.2 Penyisipan Teks Rahasia

##### 4.1.2.1 Tahapan Penyisipan Teks Rahasia

Dalam penerapan model teks steganografi dengan metode pemetaan bit, langkah-langkah penyisipan teks rahasia dapat dinyatakan pada susunan algoritma yang ditunjukkan dengan pseudocode pada Gambar 4.6.

```
(1) Get the secret text ( $M$ ) and encrypt the text ( $E$ );  
(2) Get the encrypted text ( $E$ ) and convert to its binary code equivalent ( $E_b$ ) and make 7 last bits;  
(3) Get the cover text ( $C$ ) and convert to its binary code equivalent ( $C_b$ ) and make 7 last bits;  
(4) Read sequential and repetitive bits ( $E_b$ ) left to right;  
(5) Read the bits ( $C_b$ ) sequentially per letter at 7 bits right to left and write in the stego text ( $S$ );  
(6) If bit ( $E_b$ ) = 0 finds in bit ( $C_b$ ) = 0;  
(7) Else If bit ( $E_b$ ) = 1 finds in bit ( $C_b$ ) = 1;  
(8) Then write the sequence number in the stego key ( $W$ );  
(9) Iterate steps 4-8 till the last of bits ( $E_b$ );  
(10) Send stego text ( $S$ ) and stego key ( $W$ ) to receiver.
```

Gambar 4.6 Pseudocode penyisipan teks rahasia

Adapun tahapan penyisipan teks rahasia kedalam teks sampul sebagaimana pada Gambar 4.6 adalah sebagai berikut:

1. Didapatkan teks rahasia kemudian dienkripsi menjadi teks terenkripsi;
2. Selanjutnya teks terenkripsi diubah kedalam nilai binernya dengan setiap karakter sepanjang 7 digit biner (bit);
3. Didapatkan teks sampul untuk kemudian juga diubah kedalam nilai binernya dengan setiap karakter sepanjang 7 bit;

4. Kemudian proses penyisipan dilakukan dengan membaca secara berurutan deret bit teks terenkripsi dari kiri ke kanan untuk dimasukkan kedalam deret bit teks sampul;
5. Selanjutnya deret bit teks sampul dibaca per karakter mulai karakter pertama dengan setiap 7 bit dibaca dari kanan ke kiri, untuk kemudian penyisipan tersebut ditulis dalam teks stego;
6. Dengan ketentuan jika bit teks terenkripsi yang disisipkan bernilai 0 maka pada deret bit karakter teks sampul juga harus ditemukan yang bernilai 0 yang pertama kali didapat;
7. Begitu juga untuk nilai bit 1 diberlakukan ketentuan yang sama;
8. Dengan ketentuan tersebut, didapatkan nomor urutan penyisipan teks rahasia ke dalam teks sampul untuk ditetapkan sebagai kunci stego;
9. Proses dari nomor 4 hingga nomor 8 diulang hingga seluruh deret bit teks terenkripsi masuk kedalam deret bit teks sampul;
10. Untuk selanjutnya hasil proses penyisipan berupa teks stego dan kunci stego dikirimkan ke penerima yang dituju.

#### *4.1.2.2 Hasil Uji Coba Penyisipan Teks Rahasia*

Untuk pengujian penyisipan teks rahasia dilakukan pada sebuah dokumen negara yang diambil dari salah satu surat Keputusan Presiden Republik Indonesia yaitu Keputusan Presiden Nomor 7 Tahun 2018 tentang Biaya Penyelenggaraan Ibadah Haji Tahun 2018 [26], untuk diambil sebagian isinya sejumlah 1449 karakter sebagai teks sampul sebagaimana ditunjukkan pada Gambar 4.7, dan disisipi teks rahasia yang telah terenkripsi pada Gambar 4.3. dan dihasilkan teks stego pada Gambar 4.8, dan dari proses penyisipan juga didapatkan kunci stego sebagaimana Gambar 4.9.

Menetapkan besaran Biaya Penyelenggaraan Ibadah Haji (BPIH) Tahun 1439H/2018M bagi Jemaah Haji sebagai berikut:

a. Embarkasi Aceh	sebesar Rp31.090.010,00
b. Embarkasi Medan	sebesar Rp31.840.375,00
c. Embarkasi Batam	sebesar Rp32.456.450,00
d. Embarkasi Padang	sebesar Rp33.068.245,00
e. Embarkasi Palembang	sebesar Rp33.529.675,00
f. Embarkasi Jakarta (Pondok Gede)	sebesar Rp34.532.190,00
g. Embarkasi Jakarta (Bekasi)	sebesar Rp34.532.190,00
h. Embarkasi Solo	sebesar Rp35.933.275,00
i. Embarkasi Surabaya	sebesar Rp36.091.845,00
j. Embarkasi Banjarmasin	sebesar Rp38.157.084,00
k. Embarkasi Balikpapan	sebesar Rp38.525.445,00
l. Embarkasi Makassar	sebesar Rp39.507.741,00
m. Embarkasi Lombok	sebesar Rp38.798.305,00

Menetapkan besaran BPIH Tahun 1439H/2018M bagi Tim Pemandu Haji Daerah (TPHD) sebagai berikut:

a. Embarkasi Aceh	sebesar Rp58.796.855,00
b. Embarkasi Medan	sebesar Rp59.547.220,00
c. Embarkasi Batam	sebesar Rp60.163.295,00
d. Embarkasi Padang	sebesar Rp60.775.090,00
e. Embarkasi Palembang	sebesar Rp61.236.520,00
f. Embarkasi Jakarta (Pondok Gede)	sebesar Rp62.239.035,00
g. Embarkasi Jakarta (Bekasi)	sebesar Rp62.239.035,00
h. Embarkasi Solo	sebesar Rp63.640.120,00
i. Embarkasi Surabaya	sebesar Rp63.798.690,00
j. Embarkasi Banjarmasin	sebesar Rp65.863.929,00
k. Embarkasi Balikpapan	sebesar Rp66.232.290,00
l. Embarkasi Makassar	sebesar Rp67.214.586,00
m. Embarkasi Lombok	sebesar Rp66.505.150,00

Gambar 4.7 Teks Sampul

Menetapkan besaran Biaya Penyelenggaraan Ibadah Haji (BPIH) Tahun 1439H/2018M bagi Jemaah Haji sebagai berikut:

- a. Embarkasi Aceh sebesar Rp31.090.010,00
- b. Embarkasi Medan sebesar Rp31.840.375,00
- c. Embarkasi Batam sebesar Rp32.456.450,00
- d. Embarkasi Padang sebesar Rp33.068.245,00
- e. Embarkasi Palembangsebesar Rp33.529.675,00
- f. Embarkasi Jakarta (Pondok Gede) sebesar Rp34.532.190,00
- g. Embarkasi Jakarta (Bekasi) sebesar Rp34.532.190,00
- h. Embarkasi Solo sebesar Rp35.933.275,00
- i. Embarkasi Surabaya sebesar Rp36.091.845,00
- j. Embarkasi Banjarmasin sebesar Rp38.157.084,00
- k. Embarkasi Balikpapan sebesar Rp38.525.445,00
- l. Embarkasi Makassar sebesar Rp39.507.741,00
- m. Embarkasi Lombok sebesar Rp38.798.305,00

Menetapkan besaran BPIH Tahun 1439H/2018M bagi Tim Pemandu Haji Daerah (TPHD) sebagai berikut:

- a. Embarkasi Aceh sebesar Rp58.796.855,00
- b. Embarkasi Medan sebesar Rp59.547.220,00
- c. Embarkasi Batam sebesar Rp60.163.295,00
- d. Embarkasi Padang sebesar Rp60.775.090,00
- e. Embarkasi Palembangsebesar Rp61.236.520,00
- f. Embarkasi Jakarta (Pondok Gede) sebesar Rp62.239.035,00
- g. Embarkasi Jakarta (Bekasi) sebesar Rp62.239.035,00
- h. Embarkasi Solo sebesar Rp63.640.120,00
- i. Embarkasi Surabaya sebesar Rp63.798.690,00
- j. Embarkasi Banjarmasin sebesar Rp65.863.929,00
- k. Embarkasi Balikpapan sebesar Rp66.232.290,00
- l. Embarkasi Makassar sebesar Rp67.214.586,00
- m. Embarkasi Lombok sebesar Rp66.505.150,00

Gambar 4.8 Teks Stego



```

121112511211131222622112651111122411221213213241112214151426121
121111141152426224212112111412261111122111223232441142111212126
111143121311115311511251131534213212113132612312311121126253214
35234215133314111113111111213121111211532232223253554312311111
113265232113122112261513251122121513421421211323165132221243312
111112511122122121113412421212313212131211641521516423113121231
261113111222213514412312121113161231232641112121411213216213311
12112135533124112223111111353311112261131111114235143113222121
112632112221331211221213211212132115341231122231316111222121223
111212262114121421131554432411211311112231151112412223126213111
211332155443141112231111113111113312111112512115124113113412422
21232326352213312211226113121112352355442121111221111121212114
632112613124525111621121311651111126411113111116435132111111226
211211323421411222323161321412123226252121122422155341131122212
11122112432113126152121111225315441231211111211131231211121615
251223221231544114212121231151322431111322621152111112535544114
111211211612111122133121321125211112211115543213112223231621322
311151231364231131211111625222231151115544113221121131111111326
421123113121231212512223115311113412322212111213535412121121211
121152225311341241111212126312112223121211262123211412213553321
311221111112122111112243222112615212423212131534123222121112121
313525123311131211122221122253154432311122121211111332241122122
621242113221115543114122223211135111141212111115122111211515133

```

Gambar 4.9 Kunci Stego

#### 4.1.3 Ekstraksi Teks Rahasia

##### 4.1.3.1 Tahapan Ekstraksi Teks Rahasia

Selanjutnya pada penerapan model teks steganografi dengan metode pemetaan bit dilakukan ekstraksi teks rahasia dengan langkah-langkah yang dapat dilihat pada pseudocode pada Gambar 4.10 dengan penjelasan sebagai berikut:

1. Didapatkan teks stego untuk kemudian dikonversi kedalam nilai binernya menjadi deret bit;
2. Didapatkan kunci stego untuk dibaca secara berurutan;
3. Selanjutnya deret bit teks stego dibaca per karakter (setiap 7 bit) dari kanan ke kiri;
4. Jika kunci stego bernilai 1, maka ambil bit pertama pada deret bit teks stego di setiap 7 bitnya;
5. Jika kunci stego bernilai 2, maka ambil bit kedua pada deret bit teks stego di setiap 7 bitnya;

6. Jika kunci stego bernilai 3, maka ambil bit ketiga pada deret bit teks stego di setiap 7 bitnya;
7. Jika kunci stego bernilai 4, maka ambil bit keempat pada deret bit teks stego di setiap 7 bitnya;
8. Jika kunci stego bernilai 5, maka ambil bit kelima pada deret bit teks stego di setiap 7 bitnya;
9. Jika kunci stego bernilai 6, maka ambil bit keenam pada deret bit teks stego di setiap 7 bitnya;
10. Jika kunci stego bernilai 7, maka ambil bit ketujuh pada deret bit teks stego di setiap 7 bitnya;
11. Kemudian tulis hasil bit yang didapat menjadi deret bit teks terenkripsi;
12. Ulangi proses tahapan ke 3 sampai ke 11 hingga seluruh bit pada deret bit teks stego habis;
13. Selanjutnya konversi setiap 7 bit pada deret bit teks terenkripsi kedalam karakter ASCII yang sesuai untuk ditulis menjadi teks terenkripsi;
14. Didapatkan teks terenkripsi;
15. Kemudian dilakukan proses dekripsi dari teks terenkripsi untuk didapatkan teks rahasia.

```

(1) Get the stego text ( $S$ ) and convert to its binary code
    equivalent ( $S_b$ );
(2) Get the stego key ( $W$ ) and read sequentially;
(3) Read consecutive bits ( $S_b$ ) per letter;
(4) If key ( $W$ ) = 1 then take the first bit ( $S_b$ );
(5) If key ( $W$ ) = 2 then take the second bit ( $S_b$ );
(6) If key ( $W$ ) = 3 then take the third bit ( $S_b$ );
(7) If key ( $W$ ) = 4 then take the fourth bit ( $S_b$ );
(8) If key ( $W$ ) = 5 then take the fifth bit ( $S_b$ );
(9) If key ( $W$ ) = 6 then take the sixth bit ( $S_b$ );
(10) Else If key ( $W$ ) = 7 then take the seventh bit ( $S_b$ );
(11) Then write in the file ( $E_b$ );
(12) Repeat steps 3-11 till the last of bits ( $S_b$ );
(13) Convert every 7 bits the file ( $E_b$ ) to its ASCII binary
    code equivalent and write to encrypted text ( $E$ );
(14) Obtained encrypted text ( $E$ );
(15) Decrypt ( $E$ ) to secret message ( $M$ ).

```

Gambar 4.10 Pseudocode ekstraksi teks rahasia

#### 4.1.3.2 Hasil Uji Coba Ekstraksi Teks Rahasia

Ekstraksi pada teks stego Gambar 4.8 dilakukan menggunakan kunci stego pada Gambar 4.9 dan Kunci OTP pada Gambar 4.4, dan hasil ekstraksi menunjukkan bahwa teks rahasia dapat diperoleh kembali dan utuh sebagaimana ditunjukkan pada Gambar 4.11

VisikementeriaanAgamaRepublikIndonesiaTerwujudnyaMasyarakatIndonesiaYangTaataBeragamaRukunCerdasdanSejahteraLahirBatinDalamRangkaMewujudkanIndonesiaYangBerdaulatMandiriDanBerkepribadianBerlandaskanGotongRoyong

Gambar 4.11 Teks rahasia hasil ekstraksi

## 4.2 Kapasitas Penyisipan Teks Rahasia pada Teks Sampul

Untuk mengukur kemampuan daya tampung teks sampul terhadap teks rahasia dilakukan pengukuran dengan cara membagi jumlah deret bit teks rahasia dengan deret bit teks sampul yang dapat dilihat dengan persamaan berikut

$$Capacity\ ratio = \frac{bits\ of\ secret\ text}{bits\ of\ cover\ text} \times 100 \quad (4.3)$$

Setelah dilakukan perhitungan pada teks rahasia di Gambar 4.2 terhadap teks sampul di Gambar 4.7 diketahui bahwa 1 (satu) bit teks rahasia dapat ditampung pada 7 (tujuh) bit teks sampul yang berarti 1 (satu) karakter teks sampul dapat menampung 1 (satu) bit teks rahasia dimana rasio kapasitasnya adalah sebesar 14,2%, sebagaimana perhitungan berikut:

teks rahasia = 207 karakter = 1449 bit

teks sampul = 1449 karakter = 10143 bit

maka daya tampung bit =  $7 = \frac{1449}{10143}$

dan rasio kapasitasnya adalah =  $14,2\% = \frac{1449}{10143} \times 100$

### 4.3 Imperseptibilitas Teks Stego

Untuk mengukur imperseptibilitas (kemiripan) antara teks sampul Gambar 4.7 dan teks stego Gambar 4.8 perlu dilakukan pengukuran dengan mengukur jarak kemiripannya (*similarity distance*) menggunakan *Jaro Distance* yang ditunjukkan pada Persamaan (4.4), dimana setelah dilakukan pengujian menghasilkan nilai 1 yang artinya antara teks sampul dan teks stego adalah mirip. Berikut persamaan *Jaro Distance*:

$$d_j = \frac{1}{3} \times \left( \frac{m}{|s_1|} + \frac{m}{|s_2|} + \frac{m-t}{m} \right) \quad (4.4)$$

di mana dengan ketentuan sebagai berikut:

$|s_1|$  = string ke-1 (teks sampul)

$|s_2|$  = string ke-2 (teks stego)

$m$  = adalah jumlah karakter yang sama pada  $|s_1|$  dan  $|s_2|$

$t$  = jumlah transposisi.

Nilai  $d_j$  dianggap benar jika sama dan tidak lebih dari :

$$\left\lfloor \frac{\max |s_1|, |s_2|}{2} \right\rfloor - 1 \quad (4.5)$$

Kemudian similarity distance *Jaro-Winkler Distance* didapatkan dengan persamaan berikut:

$$d_w = d_j + (lp(1 - d_j)) \quad (4.6)$$

dengan ketentuan :

$l$  = panjang prefiks umum di awal string (panjang karakter yang sama sebelum ditemukan ketidaksamaan maksimal 4)

$p$  = konstanta scaling factor bernilai 0.1.

hasil komputasi dapat dicermati dari perhingan sebagai berikut:

Teks sampul = 1449 karakter

Teks stego = 1449 karakter

maka didapatkan hasil:  $1 = \frac{1}{3} \times \left( \frac{1449}{1449} + \frac{1449}{1449} + \frac{1449-0}{1449} \right)$

Tampilan teks stego pada Gambar 4.8 dengan teks sampulnya pada Gambar 4.7 jika dicermati terlihat sedikit berbeda karena terjadi pergeseran jarak antar kata yang menggunakan Tab (Horizontal Tab), karakter Tab yang digunakan bernilai satu Tab, dimana teks sampul ditampilkan dalam format Word Processor sehingga jarak antar Tab dapat diatur sedemikian rupa tanpa merubah nilai karakter Tab yang digunakan, sedangkan teks stego yang merupakan hasil penyisipan ditampilkan dalam plain text tanpa format tampilan tertentu namun tetap bernilai sama sebagaimana teks sampulnya.

#### 4.4 Penyisipan Identifier

Untuk membedakan antara *plain text* dan *teks stego* perlu diberikan sebuah *identifier* yang berfungsi untuk mengidentifikasikan bahwa sebuah teks telah disisipi pesan rahasia yaitu dengan cara menyisipkan kode yang terdiri dari beberapa digit angka yang terdiri dari 2 digit terakhir *kunci stego* dan hasil penjumlahan dari keseluruhan *kunci stego* tersebut kemudian diletakkan diakhir teks yang disamarkan bersama kode kontributor berita jika teks dijadikan sebagai sebuah artikel berita. Hasil teks stego yang telah ditambahkan *identifier* dapat dicermati pada Gambar 4.12.

Untuk menguji tingkat kemiripan antara *plain text* dengan *teks stego* yang telah disertakan identifier maka dilakukan komputasi dengan metode *Jaro-Winkler Distance* dengan persamaan (4.4), (4.5), dan persamaan (4.6) sebagaimana berikut:

Teks sampel : 1449 karakter

Teks stego : 1461 karakter

Jaro Distance :  $0.9972 = \frac{1}{3} \times \left( \frac{1449}{1449} + \frac{1449}{1461} + \frac{1449-0}{1449} \right)$

Jaro-Winkler Distance :  $0.9983 = 0.9972 + (4.0,1(1 - 0.9972))$

Menetapkan besaran Biaya Penyelenggaraan Ibadah Haji (BPIH) Tahun 1439H/2018M bagi Jemaah Haji sebagai berikut:

- a. Embarkasi Aceh sebesar Rp31.090.010,00
- b. Embarkasi Medan sebesar Rp31.840.375,00
- c. Embarkasi Batam sebesar Rp32.456.450,00
- d. Embarkasi Padang sebesar Rp33.068.245,00
- e. Embarkasi Palembangsebesar Rp33.529.675,00
- f. Embarkasi Jakarta (Pondok Gede) sebesar Rp34.532.190,00
- g. Embarkasi Jakarta (Bekasi) sebesar Rp34.532.190,00
- h. Embarkasi Solo sebesar Rp35.933.275,00
- i. Embarkasi Surabaya sebesar Rp36.091.845,00
- j. Embarkasi Banjarmasin sebesar Rp38.157.084,00
- k. Embarkasi Balikpapan sebesar Rp38.525.445,00
- l. Embarkasi Makassar sebesar Rp39.507.741,00
- m. Embarkasi Lombok sebesar Rp38.798.305,00

Menetapkan besaran BPIH Tahun 1439H/2018M bagi Tim Pemandu Haji Daerah (TPHD) sebagai berikut:

- a. Embarkasi Aceh sebesar Rp58.796.855,00
- b. Embarkasi Medan sebesar Rp59.547.220,00
- c. Embarkasi Batam sebesar Rp60.163.295,00
- d. Embarkasi Padang sebesar Rp60.775.090,00
- e. Embarkasi Palembangsebesar Rp61.236.520,00
- f. Embarkasi Jakarta (Pondok Gede) sebesar Rp62.239.035,00
- g. Embarkasi Jakarta (Bekasi) sebesar Rp62.239.035,00
- h. Embarkasi Solo sebesar Rp63.640.120,00
- i. Embarkasi Surabaya sebesar Rp63.798.690,00
- j. Embarkasi Banjarmasin sebesar Rp65.863.929,00
- k. Embarkasi Balikpapan sebesar Rp66.232.290,00
- l. Embarkasi Makassar sebesar Rp67.214.586,00
- m. Embarkasi Lombok sebesar Rp66.505.150,00

(alp/333033)

Gambar 4.12 Teks stego yang telah disertakan identifier

#### 4.5 Analisis Teks Stego dengan OCR (*Optical Character Recognition*)

Untuk mengetahui apakah pesan rahasia yang telah disisipkan pada *teks stego* yang tercetak ataupun yang telah dikonversi dalam bentuk format baku selain *plain text* misal dalam bentuk PDF (*Portable Document Format*) atau juga dalam bentuk gambar dapat diekstrak kembali maka dilakukan pengujian menggunakan OCR (*Optical Character Recognition*).

Pengujian dilakukan pada 3 (tiga) file teks stego dengan perbedaan pada penggunaan karakter ASCII yaitu:

1. File 1 (Gambar 4.13) berupa teks stego dengan plain text dalam bentuk paragraf yang menyertakan karakter ASCII yang *non printable* yaitu karakter Tab (*Horizontal Tab*), CR (*Carriage return*) dan LF (*Line Feed*) ;
2. File 2 (Gambar 4.14) berupa teks stego dengan plain text dalam bentuk paragraf yang membentuk beberapa baris tanpa menyertakan karakter ASCII yang *non printable*;
3. File 3 (Gambar 4.15) berupa teks stego dengan plain text dalam bentuk beberapa baris yang menyertakan karakter ASCII yang *non printable* berupa karakter CR (*Carriage return*) dan LF (*Line Feed*);

Tabel 4.1 Hasil pengujian menggunakan OCR (*Optical Character Recognition*)

No.	Dokumen	Hasil Ekstraksi (%)	Imperseptibilitas	
			Jaro Distance	Jaro-Winkler Distance
1.	File 1	5,314	0.8345	0.9007
2.	File 2	42,857	0.9039	0.9423
3.	File 3	75,308	0.9617	0.9770

Hasil pengujian menggunakan OCR menunjukkan bahwa teks stego yang dihasilkan dari *plain text* dengan format baris yang meskipun menggunakan karakter *non printable* berupa CR (*Carriage return*) dan LF (*Line Feed*) seperti pada File 3, merupakan teks stego yang mempunyai skor paling tinggi dalam ekstraksi pesan rahasia dan juga dari tingkat imperseptibilitasnya, sedangkan teks stego yang dihasilkan dari *plain text* yang berupa paragraf tanpa penggalan baris pada setiap kalimatnya dan juga menyertakan karakter *non printable* berupa Tab (*Horizontal Tab*), CR (*Carriage return*) dan LF (*Line Feed*) mendapatkan skor

terendah dalam perolehan pesan rahasia dan juga imperseptibilitasnya. Hasil pengujian dapat dicermati pada Tabel 4.1.

Menetapkan besaran Biaya Penyelenggaraan Ibadah Haji (BPIH) Tahun 1439H/2018M bagi Jemaah Haji sebagai berikut:		
a.	Embarkasi Aceh	sebesar Rp31.090.010,00
b.	Embarkasi Medan	sebesar Rp31.840.375,00
c.	Embarkasi Batam	sebesar Rp32.456.450,00
d.	Embarkasi Padang	sebesar Rp33.068.245,00
e.	Embarkasi Palembang	sebesar Rp33.529.675,00
f.	Embarkasi Jakarta (Pondok Gede)	sebesar Rp34.532.190,00
g.	Embarkasi Jakarta (Bekasi)	sebesar Rp34.532.190,00
h.	Embarkasi Solo	sebesar Rp35.933.275,00
i.	Embarkasi Surabaya	sebesar Rp36.091.845,00
j.	Embarkasi Banjarmasin	sebesar Rp38.157.084,00
k.	Embarkasi Balikpapan	sebesar Rp38.525.445,00
l.	Embarkasi Makassar	sebesar Rp39.507.741,00
m.	Embarkasi Lombok	sebesar Rp38.798.305,00
Menetapkan besaran BPIH Tahun 1439H/2018M bagi Tim Pemandu Haji Daerah (TPHD) sebagai berikut:		
a.	Embarkasi Aceh	sebesar Rp58.796.855,00
b.	Embarkasi Medan	sebesar Rp59.547.220,00
c.	Embarkasi Batam	sebesar Rp60.163.295,00
d.	Embarkasi Padang	sebesar Rp60.775.090,00
e.	Embarkasi Palembang	sebesar Rp61.236.520,00
f.	Embarkasi Jakarta (Pondok Gede)	sebesar Rp62.239.035,00
g.	Embarkasi Jakarta (Bekasi)	sebesar Rp62.239.035,00
h.	Embarkasi Solo	sebesar Rp63.640.120,00
i.	Embarkasi Surabaya	sebesar Rp63.798.690,00
j.	Embarkasi Banjarmasin	sebesar Rp65.863.929,00
k.	Embarkasi Balikpapan	sebesar Rp66.232.290,00
l.	Embarkasi Makassar	sebesar Rp67.214.586,00
m.	Embarkasi Lombok	sebesar Rp66.505.150,00

Gambar 4.13 File 1

Terwujudnya Masyarakat Indonesia yang Taat Beragama, Rukun, Cerdas, dan Sejahtera Lahir Batin dalam rangka Mewujudkan Indonesia yang Berdaulat Mandiri dan Berkepribadian Berlandaskan Gotong Royong
--

Gambar 4.14 File 2



1. Meningkatkan pemahaman, dan pengamalan ajaran agama
2. Memantapkan kerukunan intra, dan antar umat beragama
3. Menyediakan pelayanan kehidupan beragama yang merata dan berkualitas
4. Meningkatkan pemanfaatan, dan kualitas pengelolaan potensi ekonomi keagamaan
5. Mewujudkan penyelenggaraan ibadah haji, dan umrah yang berkualitas dan akuntabel
6. Meningkatkan akses dan kualitas pendidikan umum berciri agama, pendidikan agama pada satuan pendidikan umum, dan pendidikan keagamaan
7. Mewujudkan tata kelola pemerintahan yang bersih, akuntabel, dan terpercaya

Gambar 4.15 File 3

*Halaman ini sengaja dikosongkan*

## BAB 5

### PENUTUP

#### 5.1 Kesimpulan

Dari proses awal hingga akhir penyelesaian penelitian berikut hal-hal yang dapat disimpulkan antara lain:

1. *Steganografi teks* dengan metode *pemetaan bit* berhasil dilakukan pada *plain text* (teks biasa) dengan cara penyisipan pada nilai binernya. Secara singkat tahapan melakukan metode ini adalah, mengenkripsi *teks rahasia* terlebih dahulu menggunakan OTP mod 26, kemudian teks yang telah terenkripsi diubah ke nilai binernya menjadi *deret bit* begitu juga teks sampulnya, selanjutnya dilakukan penyisipan *deret bit* teks rahasia kedalam *deret bit* teks sampul, dan dari proses tersebut didapatkan *teks stego* dan juga *kunci stego*.
2. Teknik enkripsi *One Time Pad* (OTP) modulo 26 yang dilakukan menghasilkan teks terenkripsi (*ciphertext*) yang random sesuai dengan kunci yang dimasukkan, *teks terenkripsi* tidak menunjukkan pola atau statistik tertentu sehingga sangat sulit untuk dipecahkan.
3. Hasil pengujian secara kuantitatif menunjukkan bahwa dengan metode pemetaan bit, *teks rahasia* berhasil ditampung dalam seluruh *teks sampul* yang terdiri dari karakter ASCII, 1 (satu) karakter teks sampul menampung 1 (satu) bit teks rahasia yang artinya satu karakter teks rahasia dapat disembunyikan dalam 7 karakter teks sampul, hal ini dapat dinyatakan bahwa rasio kapasitasnya adalah 14,2%.
4. Pengujian secara kualitatif dilakukan untuk melihat *imperseptibilitas* atau kemiripan antara teks sampul dan teks stego, yaitu dengan diukur menggunakan *similarity distance*, setelah dilakukan pengukuran didapati bahwa antara teks sampul dan stego teks sama sekali mirip atau tidak ada perbedaan baik ukuran maupun tampilannya, hal ini terbukti dengan skor didapat setelah diukur dengan *Jaro-Winkler Distance* yang bernilai 1 (satu).
5. Pengujian dengan menggunakan OCR (*Optical Character Recognition*) dilakukan untuk mengetahui kualitas *teks stego* jika dicetak ataupun dikonversi dalam bentuk selain *plain text*, dari pengujian yang dilakukan dapat

disimpulkan bahwa *teks stego* yang mempunyai susunan bentuk yang sama dengan *plain text* meskipun disertakan karakter *non printable* mendapatkan skor tertinggi baik dalam ekstraksi pesan rahasia maupun imperseptibilitasnya.

6. *Teks stego* yang dihasilkan dapat digunakan sebagai salah satu cara menjaga keamanan informasi pada *plain text*, karena teks stego dapat dikonfirmasi keutuhannya, dimana jika terjadi perubahan pada teks stego bisa diketahui dari hasil ekstraksi *teks rahasia* yang telah disisipkan sebelumnya.

## 5.2 Saran

1. Metode *Pemetaan bit* ini dapat dikembangkan pada standar pengkodean karakter selain ASCII sehingga dapat diimplementasikan pada dokumen berbahasa selain latin, misal aksara jawa, bahasa arab dan lainnya.
2. Untuk pengenkripsian pesan rahasia dapat dikembangkan dengan metode lainya untuk menambah kompleksitas dan tingkat keamanannya.
3. Sebagai kelanjutan pengembangan metode pengamanan *plain text*, perlu dicermati lagi penggunaan karakter *non printable* dan susunan kalimat *plain text* yang akan diproses menjadi *teks stego*, karena jika teks stego dicetak ataupun dikonversi dalam bentuk selain *plain text* maka tampilan hasil akhir teks stego akan sangat berpengaruh nantinya pada hasil ekstraksi pesan rahasia.

## DAFTAR PUSTAKA

- [1] Presiden Republik Indonesia Dr. H. Susilo Bambang Yudhoyono, “Undang-Undang Republik Indonesia Nomor 11 Tahun 2008 Tentang Informasi dan Transaksi Elektronik.” 2008.
- [2] Presiden Republik Indonesia Dr. H. Susilo Bambang Yudhoyono, “Peraturan Pemerintah Republik Indonesia Nomor 82 Tahun 2012 tentang Penyelenggaraan Sistem dan Transaksi Elektronik.” 2012.
- [3] Presiden Republik Indonesia Joko Widodo, “Undang-Undang Republik Indonesia Nomor 19 Tahun 2016 tentang Perubahan Atas Undang-Undang Nomor 11 Tahun 2008 Tentang Informasi dan Transaksi Elektronik.” pp. 1–14, 2016.
- [4] P. Wayner, *Disappearing cryptography : information hiding : steganography & watermarking*. Morgan Kaufmann Publishers, 2009.
- [5] M. Agarwal, “Text Steganographic Approaches: A Comparison,” *Int. J. Netw. Secur. Its Appl.*, vol. 5, no. 1, pp. 91–106, 2013.
- [6] S. Mahato, D. K. Yadav, and D. A. Khan, “Personal characters to bits mapping using Dot Pattern Character Encoding Scheme (DPCES),” *J. King Saud Univ. - Comput. Inf. Sci.*, 2017.
- [7] P. Christensson, “Plain Text Definition,” *TechTerms.*, 2010. [Online]. Available: <https://techterms.com/definition/plaintext>. [Accessed: 14-May-2017].
- [8] C. E. Mackenzie, *Coded-Character Sets: History and Development*. Addison-Wesley Longman Publishing Co., Inc., 1980.
- [9] “ASCII Conversion Chart.” [Online]. Available: [http://web.alfredstate.edu/faculty/weimandn/miscellaneous/ascii/ascii\\_index.html](http://web.alfredstate.edu/faculty/weimandn/miscellaneous/ascii/ascii_index.html).
- [10] R. Shukla, H. O. Prakash, R. P. Bhushan, S. Venkataraman, and G.

- Varadan, "Sampurna Suraksha : Unconditionally Secure And Authenticated One Time Pad Cryptosystem," 2013.
- [11] W. Stallings, *Cryptography and Network Security Principles and Practice*, vol. 139, no. 3. 2011.
  - [12] C. A. S. Wellia Shinta Sari, Eko Hari Rachmawanto, De Rosal Ignatius Moses Setiadi, "A Good Performance OTP Encryption Image based on DCT-DWT Steganography A Good Performance OTP Encryption Image based on," *Telkomnika*, vol. 15, no. January 2018, pp. 1987–1995, 2017.
  - [13] R. Munir, "Algoritma Enkripsi Citra dengan Pseudo One-Time Pad yang Menggunakan Sistem Chaos," *Konf. Nas. Inform.*, pp. 12–16, 2011.
  - [14] S. Bruce, *Applied cryptograph*, 2nd ed. John Wiley and Sons, Inc, 1996.
  - [15] B. PFITZMANN, "Information hiding terminology-results of an informal plenary meeting and additional proposals," *Proc. First Int. Work. Inf. Hiding. Springer-Verlag*, p. 347–350., 1996.
  - [16] K. Bennett, "Linguistic Steganography: Survey, Analysis, and Robustness Concerns for Hiding Information in Text," p. 30, 2004.
  - [17] M. Shirali-Shahreza and M. H. Shirali-Shahreza, "Text steganography in SMS," *2007 Int. Conf. Conver. Inf. Technol. ICCIT 2007*, pp. 2260–2265, 2007.
  - [18] W. Bender, D. Gruhl, N. Morimoto, and A. Lu, "Techniques for data hiding," *IBM Syst. J.*, vol. 35, no. 3.4, pp. 313–336, 1996.
  - [19] F. Al-Haidari, A. Gutub, K. Al-Kahsah, and J. Hamodi, "Improving security and capacity for arabic text steganography using 'Kashida' extensions," *2009 IEEE/ACS Int. Conf. Comput. Syst. Appl. AICCSA 2009*, pp. 396–399, 2009.
  - [20] W. W. Cohen, P. Ravikumar, and S. E. Fienberg, "A comparison of string metrics for matching names and records," *KDD Work. Data Clean. Object Consol.*, vol. 3, pp. 73–78, 2003.

- [21] B. Leonardo and S. Hansun, "Text documents plagiarism detection using Rabin-Karp and Jaro-Winkler distance algorithms," *Indones. J. Electr. Eng. Comput. Sci.*, vol. 5, no. 2, pp. 462–471, 2017.
- [22] F. Mohammad, J. Anarase, M. Shingote, and P. Ghanwat, "Optical Character Recognition Implementation Using Pattern Matching," *International J. Comput. Sci. Inf. Technol.*, vol. 5, no. 2, pp. 2088–2090, 2014.
- [23] N. venkat. Rao, A. s. c. . Sastry, A. s. . Chakravarthy, and K. P, "Optical character recognition technique algorithms," *J. Theor. Appl. Inf. Technol.*, vol. 1, no. 1, pp. 11–20, 2012.
- [24] S. Peffers, Ken; Tuunanen, Tuure ; Rothenberger, Marcus A.; Chatterjee, "A Design Science Research Methodology for Information Systems Research," vol. 24, no. 3, pp. 45–78, 2007.
- [25] Menteri Agama Republik indonesia Lukman Hakim Saifuddin, "Keputusan Menteri Agama RI Nomor 39 Tahun 2015," 2015.
- [26] Presiden Republik indonesia Joko Widodo, "Keputusan Presiden Republik Indonesia Nomor 7 Tahun 2018," pp. 1–3, 2018.

*Halaman ini sengaja dikosongkan*



## LAMPIRAN

### Plain Text File 2

Terwujudnya Masyarakat Indonesia yang Taat Beragama, Rukun, Cerdas, dan Sejahtera Lahir Batin dalam rangka Mewujudkan Indonesia yang Berdaulat Mandiri dan Berkepribadian Berlandaskan Gotong Royong

### Teks Rahasia File 2

KanwilKementerianAgamaAcehOk

### Plain Text File 3

1. Meningkatkan pemahaman, dan pengamalan ajaran agama
2. Memantapkan kerukunan intra, dan antar umat beragama
3. Menyediakan pelayanan kehidupan beragama yang merata dan berkualitas
4. Meningkatkan pemanfaatan, dan kualitas pengelolaan potensi ekonomi keagamaan
5. Mewujudkan penyelenggaraan ibadah haji, dan umrah yang berkualitas dan akuntabel
6. Meningkatkan akses dan kualitas pendidikan umum berciri agama, pendidikan agama pada satuan pendidikan umum, dan pendidikan keagamaan
7. Mewujudkan tata kelola pemerintahan yang bersih, akuntabel, dan terpercaya

### Teks Rahasia File

KantorWilayahKementerianAgamaProvinsiAcehSubagOrganisasiTataLaksana DanKepegawaian

### Proses Enkripsi OTP mod 26

1.	$21 + 13 = 34 / l$	36.	$8 + 17 = 25 / z$	71.	$6 + 21 = 27 / b$
2.	$8 + 9 = 17 / r$	37.	$0 + 5 = 5 / f$	72.	$19 + 21 = 40 / O$
3.	$18 + 21 = 39 / n$	38.	$19 + 20 = 39 / N$	73.	$0 + 7 = 7 / h$
4.	$8 + 5 = 13 / n$	39.	$4 + 17 = 21 / v$	74.	$0 + 5 = 5 / f$
5.	$10 + 20 = 30 / e$	40.	$17 + 20 = 37 / l$	75.	$19 + 6 = 25 / z$
6.	$4 + 8 = 12 / m$	41.	$22 + 5 = 27 / b$	76.	$1 + 6 = 7 / H$
7.	$12 + 7 = 19 / t$	42.	$20 + 6 = 26 / a$	77.	$4 + 3 = 7 / h$
8.	$4 + 0 = 4 / e$	43.	$9 + 17 = 26 / a$	78.	$17 + 5 = 22 / w$
9.	$13 + 20 = 33 / h$	44.	$20 + 4 = 24 / y$	79.	$0 + 9 = 9 / j$
10.	$19 + 17 = 36 / k$	45.	$3 + 7 = 10 / k$	80.	$6 + 3 = 9 / j$
11.	$4 + 5 = 9 / j$	46.	$13 + 5 = 18 / s$	81.	$0 + 10 = 10 / k$
12.	$17 + 9 = 26 / a$	47.	$24 + 8 = 32 / g$	82.	$12 + 18 = 30 / e$
13.	$8 + 0 = 8 / i$	48.	$0 + 0 = 0 / a$	83.	$0 + 11 = 11 / l$
14.	$0 + 13 = 13 / n$	49.	$12 + 6 = 18 / S$	84.	$17 + 0 = 17 / R$
15.	$13 + 13 = 26 / a$	50.	$0 + 7 = 7 / h$	85.	$20 + 15 = 35 / j$
16.	$0 + 5 = 5 / F$	51.	$18 + 20 = 38 / m$	86.	$10 + 14 = 24 / y$
17.	$6 + 21 = 27 / b$	52.	$24 + 5 = 29 / d$	87.	$20 + 16 = 36 / k$
18.	$0 + 20 = 20 / u$	53.	$0 + 6 = 6 / g$	88.	$13 + 8 = 21 / v$
19.	$12 + 4 = 16 / q$	54.	$17 + 9 = 26 / a$	89.	$2 + 16 = 18 / S$
20.	$0 + 7 = 7 / h$	55.	$0 + 5 = 5 / f$	90.	$4 + 22 = 26 / a$
21.	$17 + 4 = 21 / V$	56.	$10 + 13 = 23 / x$	91.	$17 + 20 = 37 / l$
22.	$4 + 16 = 20 / u$	57.	$0 + 6 = 6 / g$	92.	$3 + 8 = 11 / l$
23.	$15 + 5 = 20 / u$	58.	$19 + 7 = 26 / a$	93.	$0 + 4 = 4 / e$
24.	$20 + 9 = 29 / d$	59.	$8 + 15 = 23 / X$	94.	$18 + 20 = 38 / m$
25.	$1 + 13 = 14 / o$	60.	$13 + 8 = 21 / v$	95.	$3 + 17 = 20 / u$
26.	$11 + 21 = 32 / g$	61.	$3 + 6 = 9 / j$	96.	$0 + 24 = 24 / y$
27.	$8 + 8 = 16 / q$	62.	$14 + 9 = 23 / x$	97.	$13 + 19 = 32 / g$
28.	$10 + 5 = 15 / p$	63.	$13 + 17 = 30 / e$	98.	$18 + 19 = 37 / L$
29.	$8 + 7 = 15 / P$	64.	$4 + 4 = 8 / i$	99.	$4 + 19 = 23 / x$
30.	$13 + 21 = 34 / i$	65.	$18 + 25 = 43 / r$	100.	$9 + 6 = 15 / p$
31.	$3 + 0 = 3 / d$	66.	$8 + 12 = 20 / u$	101.	$0 + 5 = 5 / f$
32.	$14 + 4 = 18 / s$	67.	$0 + 13 = 13 / n$	102.	$7 + 6 = 13 / n$
33.	$13 + 17 = 30 / e$	68.	$24 + 23 = 47 / V$	103.	$19 + 7 = 26 / a$
34.	$4 + 9 = 13 / n$	69.	$0 + 1 = 1 / b$	104.	$4 + 5 = 9 / j$
35.	$18 + 5 = 23 / x$	70.	$13 + 2 = 15 / p$	105.	$17 + 3 = 20 / u$

106.	$0 + 9 = 9 / j$	141.	$14 + 20 = 34 / i$	176.	$17 + 24 = 41 / p$
107.	$11 + 3 = 14 / O$	142.	$13 + 19 = 32 / g$	177.	$8 + 20 = 28 / c$
108.	$0 + 10 = 10 / k$	143.	$4 + 24 = 28 / c$	178.	$1 + 19 = 20 / u$
109.	$7 + 25 = 32 / g$	144.	$18 + 19 = 37 / l$	179.	$0 + 19 = 19 / t$
110.	$8 + 12 = 20 / u$	145.	$8 + 19 = 27 / b$	180.	$3 + 22 = 25 / z$
111.	$17 + 23 = 40 / o$	146.	$0 + 16 = 16 / q$	181.	$8 + 16 = 24 / y$
112.	$1 + 13 = 14 / O$	147.	$24 + 15 = 39 / N$	182.	$0 + 22 = 22 / w$
113.	$0 + 23 = 23 / x$	148.	$0 + 15 = 15 / p$	183.	$13 + 0 = 13 / n$
114.	$19 + 1 = 20 / u$	149.	$13 + 11 = 24 / y$	184.	$1 + 3 = 4 / E$
115.	$8 + 2 = 10 / k$	150.	$6 + 10 = 16 / q$	185.	$4 + 18 = 22 / w$
116.	$13 + 21 = 34 / i$	151.	$1 + 9 = 10 / K$	186.	$17 + 5 = 22 / w$
117.	$3 + 21 = 24 / Y$	152.	$4 + 3 = 7 / h$	187.	$11 + 3 = 14 / o$
118.	$0 + 6 = 6 / g$	153.	$17 + 7 = 24 / y$	188.	$0 + 6 = 6 / g$
119.	$11 + 5 = 16 / q$	154.	$3 + 3 = 6 / g$	189.	$13 + 5 = 18 / s$
120.	$0 + 7 = 7 / h$	155.	$0 + 8 = 8 / i$	190.	$3 + 7 = 10 / k$
121.	$12 + 3 = 15 / p$	156.	$20 + 4 = 24 / y$	191.	$0 + 9 = 9 / j$
122.	$17 + 9 = 26 / A$	157.	$11 + 3 = 14 / o$	192.	$18 + 10 = 28 / c$
123.	$0 + 18 = 18 / s$	158.	$0 + 20 = 20 / u$	193.	$10 + 11 = 21 / v$
124.	$13 + 10 = 23 / x$	159.	$19 + 19 = 38 / m$	194.	$0 + 10 = 10 / k$
125.	$6 + 18 = 24 / y$	160.	$12 + 4 = 16 / Q$	195.	$13 + 11 = 24 / y$
126.	$10 + 11 = 21 / v$	161.	$0 + 19 = 19 / t$	196.	$6 + 12 = 18 / S$
127.	$0 + 0 = 0 / a$	162.	$13 + 7 = 20 / u$	197.	$14 + 13 = 27 / b$
128.	$12 + 15 = 27 / B$	163.	$3 + 21 = 24 / y$	198.	$19 + 1 = 20 / u$
129.	$4 + 16 = 20 / u$	164.	$8 + 7 = 15 / p$	199.	$14 + 21 = 35 / j$
130.	$22 + 14 = 36 / k$	165.	$17 + 21 = 38 / m$	200.	$13 + 2 = 15 / p$
131.	$20 + 16 = 36 / k$	166.	$8 + 1 = 9 / j$	201.	$6 + 23 = 29 / d$
132.	$9 + 8 = 17 / r$	167.	$3 + 2 = 5 / F$	202.	$17 + 2 = 19 / T$
133.	$20 + 22 = 42 / q$	168.	$0 + 6 = 6 / g$	203.	$14 + 21 = 35 / j$
134.	$3 + 8 = 11 / l$	169.	$13 + 2 = 15 / p$	204.	$24 + 9 = 33 / h$
135.	$10 + 20 = 30 / e$	170.	$1 + 6 = 7 / H$	205.	$14 + 13 = 27 / b$
136.	$0 + 4 = 4 / e$	171.	$4 + 15 = 19 / t$	206.	$13 + 9 = 22 / w$
137.	$13 + 22 = 35 / j$	172.	$17 + 14 = 31 / f$	207.	$6 + 3 = 9 / j$
138.	$8 + 20 = 28 / C$	173.	$10 + 8 = 18 / s$		
139.	$13 + 4 = 17 / r$	174.	$4 + 8 = 12 / m$		
140.	$3 + 17 = 20 / u$	175.	$15 + 20 = 35 / j$		

## Proses Penyisipan Pesan

1.	'0'->'1'->'1'->'1001101'->'M'	41.	'40'->'0'->'1'->'0100000'->' '
2.	'1'->'0'->'2'->'1100101'->'e'	42.	'41'->'1'->'3'->'1101100'->'l'
3.	'2'->'0'->'1'->'1101110'->'n'	43.	'42'->'1'->'2'->'1100010'->'b'
4.	'3'->'1'->'1'->'1100101'->'e'	44.	'43'->'1'->'1'->'1100001'->'a'
5.	'4'->'0'->'1'->'1110100'->'t'	45.	'44'->'1'->'3'->'1100100'->'d'
6.	'5'->'0'->'2'->'1100001'->'a'	46.	'45'->'0'->'2'->'1100001'->'a'
7.	'6'->'1'->'5'->'1110000'->'p'	47.	'46'->'1'->'4'->'1101000'->'h'
8.	'7'->'1'->'1'->'1101011'->'k'	48.	'47'->'0'->'1'->'0100000'->' '
9.	'8'->'1'->'1'->'1100001'->'a'	49.	'48'->'0'->'1'->'1001000'->'H'
10.	'9'->'1'->'2'->'1101110'->'n'	50.	'49'->'1'->'1'->'1100001'->'a'
11.	'10'->'0'->'1'->'0100000'->' '	51.	'50'->'1'->'2'->'1101010'->'j'
12.	'11'->'0'->'1'->'1100010'->'b'	52.	'51'->'0'->'2'->'1101001'->'i'
13.	'12'->'1'->'1'->'1100101'->'e'	53.	'52'->'0'->'1'->'0100000'->' '
14.	'13'->'0'->'3'->'1110011'->'s'	54.	'53'->'1'->'4'->'0101000'->'('
15.	'14'->'1'->'1'->'1100001'->'a'	55.	'54'->'0'->'1'->'1000010'->'B'
16.	'15'->'1'->'2'->'1110010'->'r'	56.	'55'->'1'->'5'->'1010000'->'P'
17.	'16'->'0'->'2'->'1100001'->'a'	57.	'56'->'1'->'1'->'1001001'->'l'
18.	'17'->'1'->'2'->'1101110'->'n'	58.	'57'->'1'->'4'->'1001000'->'H'
19.	'18'->'1'->'6'->'0100000'->' '	59.	'58'->'0'->'2'->'0101001'->')'
20.	'19'->'1'->'2'->'1000010'->'B'	60.	'59'->'1'->'6'->'0100000'->' '
21.	'20'->'0'->'2'->'1101001'->'i'	61.	'60'->'0'->'1'->'1010100'->'T'
22.	'21'->'1'->'1'->'1100001'->'a'	62.	'61'->'0'->'2'->'1100001'->'a'
23.	'22'->'1'->'1'->'1111001'->'y'	63.	'62'->'0'->'1'->'1101000'->'h'
24.	'23'->'0'->'2'->'1100001'->'a'	64.	'63'->'1'->'1'->'1110101'->'u'
25.	'24'->'1'->'6'->'0100000'->' '	65.	'64'->'1'->'2'->'1101110'->'n'
26.	'25'->'1'->'5'->'1010000'->'P'	66.	'65'->'0'->'1'->'0100000'->' '
27.	'26'->'1'->'1'->'1100101'->'e'	67.	'66'->'1'->'1'->'0110001'->'1'
28.	'27'->'0'->'1'->'1101110'->'n'	68.	'67'->'0'->'1'->'0110100'->'4'
29.	'28'->'1'->'1'->'1111001'->'y'	69.	'68'->'1'->'1'->'0110011'->'3'
30.	'29'->'1'->'1'->'1100101'->'e'	70.	'69'->'1'->'1'->'0111001'->'9'
31.	'30'->'0'->'1'->'1101100'->'l'	71.	'70'->'1'->'4'->'1001000'->'H'
32.	'31'->'0'->'2'->'1100101'->'e'	72.	'71'->'1'->'1'->'0101111'->'/'
33.	'32'->'1'->'2'->'1101110'->'n'	73.	'72'->'0'->'1'->'0110010'->'2'
34.	'33'->'0'->'4'->'1100111'->'g'	74.	'73'->'1'->'5'->'0110000'->'0'
35.	'34'->'1'->'1'->'1100111'->'g'	75.	'74'->'0'->'2'->'0110001'->'1'
36.	'35'->'1'->'1'->'1100001'->'a'	76.	'75'->'1'->'4'->'0111000'->'8'
37.	'36'->'1'->'2'->'1110010'->'r'	77.	'76'->'0'->'2'->'1001101'->'M'
38.	'37'->'0'->'2'->'1100001'->'a'	78.	'77'->'1'->'6'->'0100000'->' '
39.	'38'->'1'->'1'->'1100001'->'a'	79.	'78'->'1'->'2'->'1100010'->'b'
40.	'39'->'1'->'2'->'1101110'->'n'	80.	'79'->'0'->'2'->'1100001'->'a'

81.	'80'->'0'->'4'->'1100111'->'g'	121.	'120'->'1'->'2'->'1110010'->'r'
82.	'81'->'0'->'2'->'1101001'->'i'	122.	'121'->'1'->'1'->'1101011'->'k'
83.	'82'->'0'->'1'->'0100000'->' '	123.	'122'->'0'->'2'->'1100001'->'a'
84.	'83'->'1'->'2'->'1001010'->'j'	124.	'123'->'1'->'1'->'1110011'->'s'
85.	'84'->'1'->'1'->'1100101'->'e'	125.	'124'->'0'->'2'->'1101001'->'i'
86.	'85'->'1'->'1'->'1101101'->'m'	126.	'125'->'1'->'6'->'0100000'->' '
87.	'86'->'0'->'2'->'1100001'->'a'	127.	'126'->'1'->'1'->'1000001'->'A'
88.	'87'->'1'->'1'->'1100001'->'a'	128.	'127'->'1'->'1'->'1100011'->'c'
89.	'88'->'0'->'1'->'1101000'->'h'	129.	'128'->'1'->'1'->'1100101'->'e'
90.	'89'->'0'->'1'->'0100000'->' '	130.	'129'->'0'->'1'->'1101000'->'h'
91.	'90'->'1'->'4'->'1001000'->'H'	131.	'130'->'0'->'4'->'01001'->'
92.	'91'->'1'->'1'->'1100001'->'a'	132.	'131'->'0'->'3'->'1110011'->'s'
93.	'92'->'1'->'2'->'1101010'->'j'	133.	'132'->'1'->'1'->'1100101'->'e'
94.	'93'->'0'->'2'->'1101001'->'i'	134.	'133'->'1'->'2'->'1100010'->'b'
95.	'94'->'1'->'6'->'0100000'->' '	135.	'134'->'1'->'1'->'1100101'->'e'
96.	'95'->'1'->'1'->'1110011'->'s'	136.	'135'->'0'->'3'->'1110011'->'s'
97.	'96'->'1'->'1'->'1100101'->'e'	137.	'136'->'1'->'1'->'1100001'->'a'
98.	'97'->'0'->'1'->'1100010'->'b'	138.	'137'->'0'->'1'->'1110010'->'r'
99.	'98'->'1'->'1'->'1100001'->'a'	139.	'138'->'0'->'1'->'0100000'->' '
100.	'99'->'1'->'1'->'1100111'->'g'	140.	'139'->'0'->'1'->'1010010'->'R'
101.	'100'->'0'->'2'->'1100001'->'a'	141.	'140'->'1'->'5'->'1110000'->'p'
102.	'101'->'0'->'2'->'1101001'->'i'	142.	'141'->'0'->'3'->'0110011'->'3'
103.	'102'->'0'->'1'->'0100000'->' '	143.	'142'->'1'->'1'->'0110001'->'1'
104.	'103'->'0'->'1'->'1100010'->'b'	144.	'143'->'0'->'1'->'0101110'->'.'
105.	'104'->'1'->'1'->'1100101'->'e'	145.	'144'->'1'->'5'->'0110000'->'0'
106.	'105'->'1'->'2'->'1110010'->'r'	146.	'145'->'1'->'1'->'0111001'->'9'
107.	'106'->'0'->'2'->'1101001'->'i'	147.	'146'->'0'->'1'->'0110000'->'0'
108.	'107'->'0'->'3'->'1101011'->'k'	148.	'147'->'1'->'2'->'0101110'->'.'
109.	'108'->'0'->'2'->'1110101'->'u'	149.	'148'->'1'->'5'->'0110000'->'0'
110.	'109'->'1'->'3'->'1110100'->'t'	150.	'149'->'1'->'1'->'0110001'->'1'
111.	'110'->'1'->'2'->'0111010'->'.'	151.	'150'->'0'->'1'->'0110000'->'0'
112.	'111'->'0'->'4'->'01101'->' '	152.	'151'->'1'->'3'->'0101100'->'.'
113.	'112'->'1'->'4'->'01010'->' '	153.	'152'->'0'->'1'->'0110000'->'0'
114.	'113'->'1'->'1'->'1100001'->'a'	154.	'153'->'1'->'5'->'0110000'->'0'
115.	'114'->'0'->'1'->'0101110'->'.'	155.	'154'->'1'->'3'->'01101'->' '
116.	'115'->'0'->'4'->'01001'->'	156.	'155'->'1'->'4'->'01010'->' '
117.	'116'->'0'->'2'->'1000101'->'E'	157.	'156'->'1'->'2'->'1100010'->'b'
118.	'117'->'1'->'1'->'1101101'->'m'	158.	'157'->'0'->'1'->'0101110'->'.'
119.	'118'->'0'->'1'->'1100010'->'b'	159.	'158'->'1'->'3'->'01001'->'
120.	'119'->'1'->'1'->'1100001'->'a'	160.	'159'->'0'->'2'->'1000101'->'E'

161.	'160'->'1'->'1'->'1101101'->'m'	201.	'200'->'0'->'3'->'1100011'->'c'
162.	'161'->'1'->'2'->'1100010'->'b'	202.	'201'->'0'->'1'->'0101110'->'.'
163.	'162'->'1'->'1'->'1100001'->'a'	203.	'202'->'0'->'4'->'01001'->'
164.	'163'->'0'->'1'->'1110010'->'r'	204.	'203'->'1'->'1'->'1000101'->'E'
165.	'164'->'0'->'3'->'1101011'->'k'	205.	'204'->'1'->'1'->'1101101'->'m'
166.	'165'->'1'->'1'->'1100001'->'a'	206.	'205'->'0'->'1'->'1100010'->'b'
167.	'166'->'0'->'3'->'1110011'->'s'	207.	'206'->'1'->'1'->'1100001'->'a'
168.	'167'->'0'->'2'->'1101001'->'i'	208.	'207'->'0'->'1'->'1110010'->'r'
169.	'168'->'1'->'6'->'0100000'->' '	209.	'208'->'0'->'3'->'1101011'->'k'
170.	'169'->'1'->'1'->'1001101'->'M'	210.	'209'->'1'->'1'->'1100001'->'a'
171.	'170'->'0'->'2'->'1100101'->'e'	211.	'210'->'1'->'1'->'1110011'->'s'
172.	'171'->'1'->'3'->'1100100'->'d'	212.	'211'->'1'->'1'->'1101001'->'i'
173.	'172'->'1'->'1'->'1100001'->'a'	213.	'212'->'0'->'1'->'0100000'->' '
174.	'173'->'1'->'2'->'1101110'->'n'	214.	'213'->'0'->'1'->'1000010'->'B'
175.	'174'->'1'->'3'->'01001'->'	215.	'214'->'1'->'1'->'1100001'->'a'
176.	'175'->'1'->'1'->'1110011'->'s'	216.	'215'->'0'->'1'->'1110100'->'t'
177.	'176'->'1'->'1'->'1100101'->'e'	217.	'216'->'0'->'2'->'1100001'->'a'
178.	'177'->'0'->'1'->'1100010'->'b'	218.	'217'->'1'->'1'->'1101101'->'m'
179.	'178'->'0'->'2'->'1100101'->'e'	219.	'218'->'1'->'3'->'01001'->'
180.	'179'->'1'->'1'->'1110011'->'s'	220.	'219'->'1'->'1'->'1110011'->'s'
181.	'180'->'1'->'1'->'1100001'->'a'	221.	'220'->'0'->'2'->'1100101'->'e'
182.	'181'->'1'->'2'->'1110010'->'r'	222.	'221'->'0'->'1'->'1100010'->'b'
183.	'182'->'1'->'6'->'0100000'->' '	223.	'222'->'1'->'1'->'1100101'->'e'
184.	'183'->'1'->'2'->'1010010'->'R'	224.	'223'->'1'->'1'->'1110011'->'s'
185.	'184'->'1'->'5'->'1110000'->'p'	225.	'224'->'1'->'1'->'1100001'->'a'
186.	'185'->'0'->'3'->'0110011'->'3'	226.	'225'->'1'->'2'->'1110010'->'r'
187.	'186'->'0'->'2'->'0110001'->'1'	227.	'226'->'0'->'1'->'0100000'->' '
188.	'187'->'0'->'1'->'0101110'->'.'	228.	'227'->'0'->'1'->'1010010'->'R'
189.	'188'->'1'->'4'->'0111000'->'8'	229.	'228'->'1'->'5'->'1110000'->'p'
190.	'189'->'1'->'3'->'0110100'->'4'	230.	'229'->'0'->'3'->'0110011'->'3'
191.	'190'->'1'->'5'->'0110000'->'0'	231.	'230'->'1'->'2'->'0110010'->'2'
192.	'191'->'1'->'2'->'0101110'->'.'	232.	'231'->'1'->'2'->'0101110'->'.'
193.	'192'->'0'->'3'->'0110011'->'3'	233.	'232'->'1'->'3'->'0110100'->'4'
194.	'193'->'0'->'4'->'0110111'->'7'	234.	'233'->'0'->'2'->'0110101'->'5'
195.	'194'->'0'->'2'->'0110101'->'5'	235.	'234'->'1'->'2'->'0110110'->'6'
196.	'195'->'0'->'1'->'0101100'->','	236.	'235'->'1'->'2'->'0101110'->'.'
197.	'196'->'1'->'5'->'0110000'->'0'	237.	'236'->'1'->'3'->'0110100'->'4'
198.	'197'->'0'->'1'->'0110000'->'0'	238.	'237'->'0'->'2'->'0110101'->'5'
199.	'198'->'1'->'3'->'01101'->' '	239.	'238'->'1'->'5'->'0110000'->'0'
200.	'199'->'0'->'3'->'01010'->' '	240.	'239'->'1'->'3'->'0101100'->','

241.	'240'->'1'->'5'->'0110000'->'0'	281.	'280'->'1'->'2'->'0101110'->'.'
242.	'241'->'1'->'5'->'0110000'->'0'	282.	'281'->'1'->'2'->'0110010'->'2'
243.	'242'->'0'->'4'->'01101'->' '	283.	'282'->'0'->'1'->'0110100'->'4'
244.	'243'->'0'->'3'->'01010'->' '	284.	'283'->'0'->'2'->'0110101'->'5'
245.	'244'->'0'->'1'->'1100100'->'d'	285.	'284'->'0'->'1'->'0101100'->','
246.	'245'->'1'->'2'->'0101110'->'.'	286.	'285'->'1'->'5'->'0110000'->'0'
247.	'246'->'1'->'3'->'01001'->'	287.	'286'->'0'->'1'->'0110000'->'0'
248.	'247'->'1'->'1'->'1000101'->'E'	288.	'287'->'1'->'3'->'01101'->' '
249.	'248'->'1'->'1'->'1101101'->'m'	289.	'288'->'1'->'4'->'01010'->' '
250.	'249'->'0'->'1'->'1100010'->'b'	290.	'289'->'0'->'2'->'1100101'->'e'
251.	'250'->'1'->'1'->'1100001'->'a'	291.	'290'->'0'->'1'->'0101110'->'.'
252.	'251'->'0'->'1'->'1110010'->'r'	292.	'291'->'0'->'4'->'01001'->'
253.	'252'->'1'->'1'->'1101011'->'k'	293.	'292'->'0'->'2'->'1000101'->'E'
254.	'253'->'1'->'1'->'1100001'->'a'	294.	'293'->'1'->'1'->'1101101'->'m'
255.	'254'->'0'->'3'->'1110011'->'s'	295.	'294'->'1'->'2'->'1100010'->'b'
256.	'255'->'0'->'2'->'1101001'->'i'	296.	'295'->'1'->'1'->'1100001'->'a'
257.	'256'->'1'->'6'->'0100000'->' '	297.	'296'->'0'->'1'->'1110010'->'r'
258.	'257'->'1'->'5'->'1010000'->'P'	298.	'297'->'0'->'3'->'1101011'->'k'
259.	'258'->'0'->'2'->'1100001'->'a'	299.	'298'->'0'->'2'->'1100001'->'a'
260.	'259'->'1'->'3'->'1100100'->'d'	300.	'299'->'0'->'3'->'1110011'->'s'
261.	'260'->'0'->'2'->'1100001'->'a'	301.	'300'->'1'->'1'->'1101001'->'i'
262.	'261'->'0'->'1'->'1101110'->'n'	302.	'301'->'1'->'6'->'0100000'->' '
263.	'262'->'1'->'1'->'1100111'->'g'	303.	'302'->'1'->'5'->'1010000'->'P'
264.	'263'->'1'->'3'->'01001'->'	304.	'303'->'1'->'1'->'1100001'->'a'
265.	'264'->'1'->'1'->'1110011'->'s'	305.	'304'->'1'->'3'->'1101100'->'i'
266.	'265'->'0'->'2'->'1100101'->'e'	306.	'305'->'0'->'2'->'1100101'->'e'
267.	'266'->'1'->'2'->'1100010'->'b'	307.	'306'->'0'->'2'->'1101101'->'m'
268.	'267'->'1'->'1'->'1100101'->'e'	308.	'307'->'1'->'2'->'1100010'->'b'
269.	'268'->'1'->'1'->'1110011'->'s'	309.	'308'->'1'->'1'->'1100001'->'a'
270.	'269'->'0'->'2'->'1100001'->'a'	310.	'309'->'1'->'2'->'1101110'->'n'
271.	'270'->'1'->'2'->'1110010'->'r'	311.	'310'->'0'->'4'->'1100111'->'g'
272.	'271'->'1'->'6'->'0100000'->' '	312.	'311'->'1'->'3'->'01001'->'
273.	'272'->'0'->'1'->'1010010'->'R'	313.	'312'->'0'->'3'->'1110011'->'s'
274.	'273'->'1'->'5'->'1110000'->'p'	314.	'313'->'1'->'1'->'1100101'->'e'
275.	'274'->'1'->'1'->'0110011'->'3'	315.	'314'->'1'->'2'->'1100010'->'b'
276.	'275'->'0'->'3'->'0110011'->'3'	316.	'315'->'1'->'1'->'1100101'->'e'
277.	'276'->'1'->'2'->'0101110'->'.'	317.	'316'->'1'->'1'->'1110011'->'s'
278.	'277'->'1'->'5'->'0110000'->'0'	318.	'317'->'1'->'1'->'1100001'->'a'
279.	'278'->'0'->'1'->'0110110'->'6'	319.	'318'->'0'->'1'->'1110010'->'r'
280.	'279'->'0'->'1'->'0111000'->'8'	320.	'319'->'0'->'1'->'0100000'->' '

321.	'320'->'1'->'2'->'1010010'->'R'	361.	'360'->'0'->'5'->'1101111'->'o'
322.	'321'->'1'->'5'->'1110000'->'p'	362.	'361'->'1'->'2'->'1101110'->'n'
323.	'322'->'1'->'1'->'0110011'->'3'	363.	'362'->'0'->'1'->'1100100'->'d'
324.	'323'->'1'->'1'->'0110011'->'3'	364.	'363'->'0'->'5'->'1101111'->'o'
325.	'324'->'0'->'1'->'0101110'->','	365.	'364'->'1'->'1'->'1101011'->'k'
326.	'325'->'0'->'2'->'0110101'->'5'	366.	'365'->'1'->'6'->'0100000'->' '
327.	'326'->'1'->'2'->'0110010'->'2'	367.	'366'->'0'->'4'->'1000111'->'G'
328.	'327'->'1'->'1'->'0111001'->'9'	368.	'367'->'0'->'2'->'1100101'->'e'
329.	'328'->'1'->'2'->'0101110'->','	369.	'368'->'1'->'3'->'1100100'->'d'
330.	'329'->'1'->'2'->'0110110'->'6'	370.	'369'->'1'->'1'->'1100101'->'e'
331.	'330'->'1'->'1'->'0110111'->'7'	371.	'370'->'1'->'1'->'0101001'->')'
332.	'331'->'0'->'2'->'0110101'->'5'	372.	'371'->'1'->'3'->'01001'->'
333.	'332'->'0'->'1'->'0101100'->','	373.	'372'->'1'->'1'->'1110011'->'s'
334.	'333'->'0'->'1'->'0110000'->'0'	374.	'373'->'0'->'2'->'1100101'->'e'
335.	'334'->'0'->'1'->'0110000'->'0'	375.	'374'->'0'->'1'->'1100010'->'b'
336.	'335'->'1'->'3'->'01101'->' '	376.	'375'->'0'->'2'->'1100101'->'e'
337.	'336'->'1'->'4'->'01010'->' '	377.	'376'->'0'->'3'->'1110011'->'s'
338.	'337'->'0'->'1'->'1100110'->'f'	378.	'377'->'1'->'1'->'1100001'->'a'
339.	'338'->'1'->'2'->'0101110'->','	379.	'378'->'1'->'2'->'1110010'->'r'
340.	'339'->'0'->'4'->'01001'->'	380.	'379'->'1'->'6'->'0100000'->' '
341.	'340'->'0'->'2'->'1000101'->'E'	381.	'380'->'0'->'1'->'1010010'->'R'
342.	'341'->'1'->'1'->'1101101'->'m'	382.	'381'->'0'->'1'->'1110000'->'p'
343.	'342'->'1'->'2'->'1100010'->'b'	383.	'382'->'1'->'1'->'0110011'->'3'
344.	'343'->'1'->'1'->'1100001'->'a'	384.	'383'->'1'->'3'->'0110100'->'4'
345.	'344'->'1'->'2'->'1110010'->'r'	385.	'384'->'0'->'1'->'0101110'->','
346.	'345'->'0'->'3'->'1101011'->'k'	386.	'385'->'1'->'1'->'0110101'->'5'
347.	'346'->'1'->'1'->'1100001'->'a'	387.	'386'->'1'->'1'->'0110011'->'3'
348.	'347'->'0'->'3'->'1110011'->'s'	388.	'387'->'1'->'2'->'0110010'->'2'
349.	'348'->'0'->'2'->'1101001'->'i'	389.	'388'->'1'->'2'->'0101110'->','
350.	'349'->'0'->'1'->'0100000'->' '	390.	'389'->'0'->'2'->'0110001'->'1'
351.	'350'->'1'->'2'->'1001010'->'J'	391.	'390'->'0'->'2'->'0111001'->'9'
352.	'351'->'1'->'1'->'1100001'->'a'	392.	'391'->'0'->'1'->'0110000'->'0'
353.	'352'->'0'->'3'->'1101011'->'k'	393.	'392'->'1'->'3'->'0101100'->','
354.	'353'->'1'->'1'->'1100001'->'a'	394.	'393'->'1'->'5'->'0110000'->'0'
355.	'354'->'1'->'2'->'1110010'->'r'	395.	'394'->'0'->'1'->'0110000'->'0'
356.	'355'->'0'->'1'->'1110100'->'t'	396.	'395'->'0'->'4'->'01101'->' '
357.	'356'->'1'->'1'->'1100001'->'a'	397.	'396'->'1'->'4'->'01010'->' '
358.	'357'->'1'->'6'->'0100000'->' '	398.	'397'->'1'->'1'->'1100111'->'g'
359.	'358'->'1'->'4'->'0101000'->'('	399.	'398'->'1'->'2'->'0101110'->','
360.	'359'->'0'->'1'->'1010000'->'P'	400.	'399'->'1'->'3'->'01001'->'



401.	'400'->'1'->'1'->'1000101'->'E'	441.	'440'->'1'->'1'->'0110101'->'5'
402.	'401'->'0'->'2'->'1101101'->'m'	442.	'441'->'1'->'1'->'0110011'->'3'
403.	'402'->'0'->'1'->'1100010'->'b'	443.	'442'->'1'->'2'->'0110010'->'2'
404.	'403'->'0'->'2'->'1100001'->'a'	444.	'443'->'0'->'1'->'0101110'->'.'
405.	'404'->'0'->'1'->'1110010'->'r'	445.	'444'->'1'->'1'->'0110001'->'1'
406.	'405'->'1'->'1'->'1101011'->'k'	446.	'445'->'0'->'2'->'0111001'->'9'
407.	'406'->'1'->'1'->'1100001'->'a'	447.	'446'->'0'->'1'->'0110000'->'0'
408.	'407'->'0'->'3'->'1110011'->'s'	448.	'447'->'1'->'3'->'0101100'->'.'
409.	'408'->'1'->'1'->'1101001'->'i'	449.	'448'->'1'->'5'->'0110000'->'0'
410.	'409'->'1'->'6'->'0100000'->' '	450.	'449'->'1'->'5'->'0110000'->'0'
411.	'410'->'0'->'1'->'1001010'->'j'	451.	'450'->'1'->'3'->'01101'->' '
412.	'411'->'0'->'2'->'1100001'->'a'	452.	'451'->'0'->'3'->'01010'->' '
413.	'412'->'0'->'3'->'1101011'->'k'	453.	'452'->'0'->'1'->'1101000'->'h'
414.	'413'->'1'->'1'->'1100001'->'a'	454.	'453'->'1'->'2'->'0101110'->'.'
415.	'414'->'1'->'2'->'1110010'->'r'	455.	'454'->'0'->'4'->'01001'->' '
416.	'415'->'1'->'3'->'1110100'->'t'	456.	'455'->'1'->'1'->'1000101'->'E'
417.	'416'->'0'->'2'->'1100001'->'a'	457.	'456'->'1'->'1'->'1101101'->'m'
418.	'417'->'1'->'6'->'0100000'->' '	458.	'457'->'1'->'2'->'1100010'->'b'
419.	'418'->'1'->'4'->'0101000'->'(')	459.	'458'->'0'->'2'->'1100001'->'a'
420.	'419'->'0'->'1'->'1000010'->'B'	460.	'459'->'1'->'2'->'1110010'->'r'
421.	'420'->'1'->'1'->'1100101'->'e'	461.	'460'->'0'->'3'->'1101011'->'k'
422.	'421'->'1'->'1'->'1101011'->'k'	462.	'461'->'1'->'1'->'1100001'->'a'
423.	'422'->'0'->'2'->'1100001'->'a'	463.	'462'->'1'->'1'->'1110011'->'s'
424.	'423'->'1'->'1'->'1110011'->'s'	464.	'463'->'1'->'1'->'1101001'->'i'
425.	'424'->'0'->'2'->'1101001'->'i'	465.	'464'->'0'->'1'->'0100000'->' '
426.	'425'->'1'->'1'->'0101001'->'j'	466.	'465'->'1'->'1'->'1010011'->'S'
427.	'426'->'0'->'4'->'01001'->' '	467.	'466'->'1'->'1'->'1101111'->'o'
428.	'427'->'1'->'1'->'1110011'->'s'	468.	'467'->'1'->'3'->'1101100'->'l'
429.	'428'->'1'->'1'->'1100101'->'e'	469.	'468'->'0'->'5'->'1101111'->'o'
430.	'429'->'1'->'2'->'1100010'->'b'	470.	'469'->'1'->'3'->'01001'->' '
431.	'430'->'1'->'1'->'1100101'->'e'	471.	'470'->'0'->'3'->'1110011'->'s'
432.	'431'->'0'->'3'->'1110011'->'s'	472.	'471'->'1'->'1'->'1100101'->'e'
433.	'432'->'0'->'2'->'1100001'->'a'	473.	'472'->'0'->'1'->'1100010'->'b'
434.	'433'->'0'->'1'->'1110010'->'r'	474.	'473'->'1'->'1'->'1100101'->'e'
435.	'434'->'1'->'6'->'0100000'->' '	475.	'474'->'1'->'1'->'1110011'->'s'
436.	'435'->'1'->'2'->'1010010'->'R'	476.	'475'->'0'->'2'->'1100001'->'a'
437.	'436'->'0'->'1'->'1110000'->'p'	477.	'476'->'1'->'2'->'1110010'->'r'
438.	'437'->'0'->'3'->'0110011'->'3'	478.	'477'->'1'->'6'->'0100000'->' '
439.	'438'->'1'->'3'->'0110100'->'4'	479.	'478'->'0'->'1'->'1010010'->'R'
440.	'439'->'0'->'1'->'0101110'->'.'	480.	'479'->'0'->'1'->'1110000'->'p'

481.	'480'->'0'->'3'->'0110011'->'3'	521.	'520'->'1'->'1'->'1100101'->'e'
482.	'481'->'1'->'1'->'0110101'->'5'	522.	'521'->'1'->'1'->'1110011'->'s'
483.	'482'->'0'->'1'->'0101110'->'.'	523.	'522'->'0'->'2'->'1100001'->'a'
484.	'483'->'1'->'1'->'0111001'->'9'	524.	'523'->'1'->'2'->'1110010'->'r'
485.	'484'->'1'->'1'->'0110011'->'3'	525.	'524'->'0'->'1'->'0100000'->' '
486.	'485'->'1'->'1'->'0110011'->'3'	526.	'525'->'1'->'2'->'1010010'->'R'
487.	'486'->'0'->'1'->'0101110'->'.'	527.	'526'->'0'->'1'->'1110000'->'p'
488.	'487'->'0'->'1'->'0110010'->'2'	528.	'527'->'0'->'3'->'0110011'->'3'
489.	'488'->'0'->'4'->'0110111'->'7'	529.	'528'->'1'->'2'->'0110110'->'6'
490.	'489'->'0'->'2'->'0110101'->'5'	530.	'529'->'0'->'1'->'0101110'->'.'
491.	'490'->'1'->'3'->'0101100'->'.'	531.	'530'->'0'->'1'->'0110000'->'0'
492.	'491'->'1'->'5'->'0110000'->'0'	532.	'531'->'0'->'2'->'0111001'->'9'
493.	'492'->'0'->'1'->'0110000'->'0'	533.	'532'->'1'->'1'->'0110001'->'1'
494.	'493'->'0'->'4'->'01101'->' '	534.	'533'->'1'->'2'->'0101110'->'.'
495.	'494'->'0'->'3'->'01010'->' '	535.	'534'->'0'->'1'->'0111000'->'8'
496.	'495'->'1'->'1'->'1101001'->'i'	536.	'535'->'1'->'3'->'0110100'->'4'
497.	'496'->'0'->'1'->'0101110'->'.'	537.	'536'->'0'->'2'->'0110101'->'5'
498.	'497'->'1'->'3'->'01001'->'	538.	'537'->'0'->'1'->'0101100'->'.'
499.	'498'->'0'->'2'->'1000101'->'E'	539.	'538'->'0'->'1'->'0110000'->'0'
500.	'499'->'0'->'2'->'1101101'->'m'	540.	'539'->'1'->'5'->'0110000'->'0'
501.	'500'->'1'->'2'->'1100010'->'b'	541.	'540'->'1'->'3'->'01101'->' '
502.	'501'->'1'->'1'->'1100001'->'a'	542.	'541'->'1'->'4'->'01010'->' '
503.	'502'->'1'->'2'->'1110010'->'r'	543.	'542'->'0'->'1'->'1101010'->'j'
504.	'503'->'1'->'1'->'1101011'->'k'	544.	'543'->'1'->'2'->'0101110'->'.'
505.	'504'->'1'->'1'->'1100001'->'a'	545.	'544'->'1'->'3'->'01001'->'
506.	'505'->'1'->'1'->'1110011'->'s'	546.	'545'->'1'->'1'->'1000101'->'E'
507.	'506'->'0'->'2'->'1101001'->'i'	547.	'546'->'1'->'1'->'1101101'->'m'
508.	'507'->'1'->'6'->'0100000'->' '	548.	'547'->'1'->'2'->'1100010'->'b'
509.	'508'->'0'->'3'->'1010011'->'S'	549.	'548'->'0'->'2'->'1100001'->'a'
510.	'509'->'0'->'2'->'1110101'->'u'	550.	'549'->'1'->'2'->'1110010'->'r'
511.	'510'->'0'->'1'->'1110010'->'r'	551.	'550'->'0'->'3'->'1101011'->'k'
512.	'511'->'1'->'1'->'1100001'->'a'	552.	'551'->'1'->'1'->'1100001'->'a'
513.	'512'->'1'->'2'->'1100010'->'b'	553.	'552'->'0'->'3'->'1110011'->'s'
514.	'513'->'0'->'2'->'1100001'->'a'	554.	'553'->'1'->'1'->'1101001'->'i'
515.	'514'->'0'->'2'->'1111001'->'y'	555.	'554'->'1'->'6'->'0100000'->' '
516.	'515'->'1'->'1'->'1100001'->'a'	556.	'555'->'0'->'1'->'1000010'->'B'
517.	'516'->'1'->'3'->'01001'->'	557.	'556'->'1'->'1'->'1100001'->'a'
518.	'517'->'0'->'3'->'1110011'->'s'	558.	'557'->'0'->'1'->'1101110'->'n'
519.	'518'->'1'->'1'->'1100101'->'e'	559.	'558'->'1'->'2'->'1101010'->'j'
520.	'519'->'1'->'2'->'1100010'->'b'	560.	'559'->'0'->'2'->'1100001'->'a'

561.	'560'->'1'->'2'->'1110010'->'r'	601.	'600'->'0'->'3'->'1101011'->'k'
562.	'561'->'1'->'1'->'1101101'->'m'	602.	'601'->'1'->'1'->'1100001'->'a'
563.	'562'->'0'->'2'->'1100001'->'a'	603.	'602'->'1'->'1'->'1110011'->'s'
564.	'563'->'1'->'1'->'1110011'->'s'	604.	'603'->'1'->'1'->'1101001'->'i'
565.	'564'->'0'->'2'->'1101001'->'i'	605.	'604'->'0'->'1'->'0100000'->' '
566.	'565'->'1'->'2'->'1101110'->'n'	606.	'605'->'1'->'2'->'1000010'->'B'
567.	'566'->'1'->'3'->'01001'->'	607.	'606'->'0'->'2'->'1100001'->'a'
568.	'567'->'1'->'1'->'1110011'->'s'	608.	'607'->'1'->'3'->'1101100'->'l'
569.	'568'->'1'->'1'->'1100101'->'e'	609.	'608'->'1'->'1'->'1101001'->'i'
570.	'569'->'0'->'1'->'1100010'->'b'	610.	'609'->'1'->'1'->'1101011'->'k'
571.	'570'->'0'->'2'->'1100101'->'e'	611.	'610'->'1'->'5'->'1110000'->'p'
572.	'571'->'1'->'1'->'1110011'->'s'	612.	'611'->'1'->'1'->'1100001'->'a'
573.	'572'->'0'->'2'->'1100001'->'a'	613.	'612'->'0'->'1'->'1110000'->'p'
574.	'573'->'1'->'2'->'1110010'->'r'	614.	'613'->'1'->'1'->'1100001'->'a'
575.	'574'->'1'->'6'->'0100000'->' '	615.	'614'->'1'->'2'->'1101110'->'n'
576.	'575'->'1'->'2'->'1010010'->'R'	616.	'615'->'0'->'4'->'01001'->'
577.	'576'->'0'->'1'->'1110000'->'p'	617.	'616'->'1'->'1'->'1110011'->'s'
578.	'577'->'1'->'1'->'0110011'->'3'	618.	'617'->'0'->'2'->'1100101'->'e'
579.	'578'->'1'->'4'->'0111000'->'8'	619.	'618'->'1'->'2'->'1100010'->'b'
580.	'579'->'0'->'1'->'0101110'->'.'	620.	'619'->'0'->'2'->'1100101'->'e'
581.	'580'->'0'->'2'->'0110001'->'1'	621.	'620'->'0'->'3'->'1110011'->'s'
582.	'581'->'1'->'1'->'0110101'->'5'	622.	'621'->'1'->'1'->'1100001'->'a'
583.	'582'->'0'->'4'->'0110111'->'7'	623.	'622'->'1'->'2'->'1110010'->'r'
584.	'583'->'1'->'2'->'0101110'->'.'	624.	'623'->'1'->'6'->'0100000'->' '
585.	'584'->'0'->'1'->'0110000'->'0'	625.	'624'->'1'->'2'->'1010010'->'R'
586.	'585'->'0'->'1'->'0111000'->'8'	626.	'625'->'0'->'1'->'1110000'->'p'
587.	'586'->'1'->'3'->'0110100'->'4'	627.	'626'->'0'->'3'->'0110011'->'3'
588.	'587'->'0'->'1'->'0101100'->','	628.	'627'->'0'->'1'->'0111000'->'8'
589.	'588'->'1'->'5'->'0110000'->'0'	629.	'628'->'0'->'1'->'0101110'->'.'
590.	'589'->'1'->'5'->'0110000'->'0'	630.	'629'->'1'->'1'->'0110101'->'5'
591.	'590'->'0'->'4'->'01101'->' '	631.	'630'->'1'->'2'->'0110010'->'2'
592.	'591'->'1'->'4'->'01010'->' '	632.	'631'->'1'->'1'->'0110101'->'5'
593.	'592'->'0'->'3'->'1101011'->'k'	633.	'632'->'0'->'1'->'0101110'->'.'
594.	'593'->'1'->'2'->'0101110'->'.'	634.	'633'->'1'->'3'->'0110100'->'4'
595.	'594'->'0'->'4'->'01001'->'	635.	'634'->'1'->'3'->'0110100'->'4'
596.	'595'->'1'->'1'->'1000101'->'E'	636.	'635'->'0'->'2'->'0110101'->'5'
597.	'596'->'1'->'1'->'1101101'->'m'	637.	'636'->'0'->'1'->'0101100'->','
598.	'597'->'1'->'2'->'1100010'->'b'	638.	'637'->'1'->'5'->'0110000'->'0'
599.	'598'->'1'->'1'->'1100001'->'a'	639.	'638'->'1'->'5'->'0110000'->'0'
600.	'599'->'0'->'1'->'1110010'->'r'	640.	'639'->'0'->'4'->'01101'->' '

641.	'640'->'1'->'4'->'01010'->' '	681.	'680'->'0'->'4'->'0110111'->'7'
642.	'641'->'1'->'3'->'1101100'->'l'	682.	'681'->'0'->'1'->'0110100'->'4'
643.	'642'->'0'->'1'->'0101110'->'.'	683.	'682'->'1'->'1'->'0110001'->'1'
644.	'643'->'0'->'4'->'01001'->'	684.	'683'->'1'->'3'->'0101100'->'.'
645.	'644'->'1'->'1'->'1000101'->'E'	685.	'684'->'0'->'1'->'0110000'->'0'
646.	'645'->'1'->'1'->'1101101'->'m'	686.	'685'->'0'->'1'->'0110000'->'0'
647.	'646'->'0'->'1'->'1100010'->'b'	687.	'686'->'1'->'3'->'01101'->' '
648.	'647'->'0'->'2'->'1100001'->'a'	688.	'687'->'1'->'4'->'01010'->' '
649.	'648'->'1'->'2'->'1110010'->'r'	689.	'688'->'1'->'1'->'1101101'->'m'
650.	'649'->'0'->'3'->'1101011'->'k'	690.	'689'->'1'->'2'->'0101110'->'.'
651.	'650'->'1'->'1'->'1100001'->'a'	691.	'690'->'0'->'4'->'01001'->'
652.	'651'->'1'->'1'->'1110011'->'s'	692.	'691'->'0'->'2'->'1000101'->'E'
653.	'652'->'1'->'1'->'1101001'->'i'	693.	'692'->'0'->'2'->'1101101'->'m'
654.	'653'->'0'->'1'->'0100000'->' '	694.	'693'->'1'->'2'->'1100010'->'b'
655.	'654'->'1'->'1'->'1001101'->'M'	695.	'694'->'1'->'1'->'1100001'->'a'
656.	'655'->'1'->'1'->'1100001'->'a'	696.	'695'->'1'->'2'->'1110010'->'r'
657.	'656'->'0'->'3'->'1101011'->'k'	697.	'696'->'0'->'3'->'1101011'->'k'
658.	'657'->'1'->'1'->'1100001'->'a'	698.	'697'->'0'->'2'->'1100001'->'a'
659.	'658'->'1'->'1'->'1110011'->'s'	699.	'698'->'0'->'3'->'1110011'->'s'
660.	'659'->'1'->'1'->'1110011'->'s'	700.	'699'->'0'->'2'->'1101001'->'i'
661.	'660'->'1'->'1'->'1100001'->'a'	701.	'700'->'1'->'6'->'0100000'->' '
662.	'661'->'0'->'1'->'1110010'->'r'	702.	'701'->'1'->'3'->'1001100'->'L'
663.	'662'->'1'->'3'->'01001'->'	703.	'702'->'0'->'5'->'1101111'->'o'
664.	'663'->'0'->'3'->'1110011'->'s'	704.	'703'->'0'->'2'->'1101101'->'m'
665.	'664'->'1'->'1'->'1100101'->'e'	705.	'704'->'1'->'2'->'1100010'->'b'
666.	'665'->'1'->'2'->'1100010'->'b'	706.	'705'->'1'->'1'->'1101111'->'o'
667.	'666'->'1'->'1'->'1100101'->'e'	707.	'706'->'0'->'3'->'1101011'->'k'
668.	'667'->'1'->'1'->'1110011'->'s'	708.	'707'->'1'->'3'->'01001'->'
669.	'668'->'1'->'1'->'1100001'->'a'	709.	'708'->'1'->'1'->'1110011'->'s'
670.	'669'->'0'->'1'->'1110010'->'r'	710.	'709'->'0'->'2'->'1100101'->'e'
671.	'670'->'0'->'1'->'0100000'->' '	711.	'710'->'1'->'2'->'1100010'->'b'
672.	'671'->'1'->'2'->'1010010'->'R'	712.	'711'->'1'->'1'->'1100101'->'e'
673.	'672'->'1'->'5'->'1110000'->'p'	713.	'712'->'1'->'1'->'1110011'->'s'
674.	'673'->'1'->'1'->'0110011'->'3'	714.	'713'->'0'->'2'->'1100001'->'a'
675.	'674'->'0'->'2'->'0111001'->'9'	715.	'714'->'1'->'2'->'1110010'->'r'
676.	'675'->'0'->'1'->'0101110'->'.'	716.	'715'->'1'->'6'->'0100000'->' '
677.	'676'->'1'->'1'->'0110101'->'5'	717.	'716'->'0'->'1'->'1010010'->'R'
678.	'677'->'1'->'5'->'0110000'->'0'	718.	'717'->'0'->'1'->'1110000'->'p'
679.	'678'->'1'->'1'->'0110111'->'7'	719.	'718'->'0'->'3'->'0110011'->'3'
680.	'679'->'1'->'2'->'0101110'->'.'	720.	'719'->'0'->'1'->'0111000'->'8'

721.	'720'->'1'->'2'->'0101110'->'.'	761.	'760'->'1'->'1'->'1110101'->'u'
722.	'721'->'1'->'1'->'0110111'->'7'	762.	'761'->'1'->'2'->'1101110'->'n'
723.	'722'->'1'->'1'->'0111001'->'9'	763.	'762'->'1'->'6'->'0100000'->' '
724.	'723'->'0'->'1'->'0111000'->'8'	764.	'763'->'1'->'1'->'0110001'->'1'
725.	'724'->'1'->'2'->'0101110'->'.'	765.	'764'->'1'->'3'->'0110100'->'4'
726.	'725'->'0'->'3'->'0110011'->'3'	766.	'765'->'1'->'1'->'0110011'->'3'
727.	'726'->'1'->'5'->'0110000'->'0'	767.	'766'->'0'->'2'->'0111001'->'9'
728.	'727'->'0'->'2'->'0110101'->'5'	768.	'767'->'1'->'4'->'1001000'->'H'
729.	'728'->'1'->'3'->'0101100'->','	769.	'768'->'0'->'5'->'0101111'->'/'
730.	'729'->'1'->'5'->'0110000'->'0'	770.	'769'->'1'->'2'->'0110010'->'2'
731.	'730'->'1'->'5'->'0110000'->'0'	771.	'770'->'1'->'5'->'0110000'->'0'
732.	'731'->'0'->'4'->'01101'->' '	772.	'771'->'1'->'1'->'0110001'->'1'
733.	'732'->'1'->'4'->'01010'->' '	773.	'772'->'0'->'1'->'0111000'->'8'
734.	'733'->'0'->'2'->'1001101'->'M'	774.	'773'->'1'->'1'->'1001101'->'M'
735.	'734'->'1'->'1'->'1100101'->'e'	775.	'774'->'1'->'6'->'0100000'->' '
736.	'735'->'1'->'2'->'1101110'->'n'	776.	'775'->'1'->'2'->'1100010'->'b'
737.	'736'->'1'->'1'->'1100101'->'e'	777.	'776'->'1'->'1'->'1100001'->'a'
738.	'737'->'0'->'1'->'1110100'->'t'	778.	'777'->'1'->'1'->'1100111'->'g'
739.	'738'->'1'->'1'->'1100001'->'a'	779.	'778'->'0'->'2'->'1101001'->'i'
740.	'739'->'0'->'1'->'1110000'->'p'	780.	'779'->'0'->'1'->'0100000'->' '
741.	'740'->'1'->'1'->'1101011'->'k'	781.	'780'->'1'->'3'->'1010100'->'T'
742.	'741'->'0'->'2'->'1100001'->'a'	782.	'781'->'1'->'1'->'1101001'->'i'
743.	'742'->'1'->'2'->'1101110'->'n'	783.	'782'->'1'->'1'->'1101101'->'m'
744.	'743'->'0'->'1'->'0100000'->' '	784.	'783'->'1'->'6'->'0100000'->' '
745.	'744'->'0'->'1'->'1100010'->'b'	785.	'784'->'1'->'5'->'1010000'->'P'
746.	'745'->'1'->'1'->'1100101'->'e'	786.	'785'->'1'->'1'->'1100101'->'e'
747.	'746'->'1'->'1'->'1110011'->'s'	787.	'786'->'1'->'1'->'1101101'->'m'
748.	'747'->'1'->'1'->'1100001'->'a'	788.	'787'->'1'->'1'->'1100001'->'a'
749.	'748'->'1'->'2'->'1110010'->'r'	789.	'788'->'0'->'1'->'1101110'->'n'
750.	'749'->'1'->'1'->'1100001'->'a'	790.	'789'->'0'->'1'->'1100100'->'d'
751.	'750'->'1'->'2'->'1101110'->'n'	791.	'790'->'0'->'2'->'1110101'->'u'
752.	'751'->'0'->'1'->'0100000'->' '	792.	'791'->'1'->'6'->'0100000'->' '
753.	'752'->'1'->'2'->'1000010'->'B'	793.	'792'->'1'->'4'->'1001000'->'H'
754.	'753'->'0'->'1'->'1010000'->'P'	794.	'793'->'1'->'1'->'1100001'->'a'
755.	'754'->'1'->'1'->'1001001'->'l'	795.	'794'->'0'->'1'->'1101010'->'j'
756.	'755'->'1'->'4'->'1001000'->'H'	796.	'795'->'1'->'1'->'1101001'->'i'
757.	'756'->'1'->'6'->'0100000'->' '	797.	'796'->'0'->'1'->'0100000'->' '
758.	'757'->'1'->'3'->'1010100'->'T'	798.	'797'->'1'->'3'->'1000100'->'D'
759.	'758'->'0'->'2'->'1100001'->'a'	799.	'798'->'1'->'1'->'1100001'->'a'
760.	'759'->'0'->'1'->'1101000'->'h'	800.	'799'->'1'->'1'->'1100101'->'e'

801.	'800'->'0'->'1'->'1110010'->'r'	841.	'840'->'1'->'1'->'1101001'->'i'
802.	'801'->'1'->'1'->'1100001'->'a'	842.	'841'->'1'->'6'->'0100000'->' '
803.	'802'->'0'->'1'->'1101000'->'h'	843.	'842'->'1'->'1'->'1000001'->'A'
804.	'803'->'1'->'6'->'0100000'->' '	844.	'843'->'0'->'3'->'1100011'->'c'
805.	'804'->'1'->'4'->'0101000'->'('	845.	'844'->'0'->'2'->'1100101'->'e'
806.	'805'->'1'->'3'->'1010100'->'T'	846.	'845'->'0'->'1'->'1101000'->'h'
807.	'806'->'1'->'5'->'1010000'->'P'	847.	'846'->'0'->'4'->'01001'->'
808.	'807'->'0'->'1'->'1001000'->'H'	848.	'847'->'1'->'1'->'1110011'->'s'
809.	'808'->'1'->'3'->'1000100'->'D'	849.	'848'->'0'->'2'->'1100101'->'e'
810.	'809'->'0'->'2'->'0101001'->')'	850.	'849'->'0'->'1'->'1100010'->'b'
811.	'810'->'0'->'1'->'0100000'->' '	851.	'850'->'0'->'2'->'1100101'->'e'
812.	'811'->'1'->'1'->'1110011'->'s'	852.	'851'->'0'->'3'->'1110011'->'s'
813.	'812'->'1'->'1'->'1100101'->'e'	853.	'852'->'0'->'2'->'1100001'->'a'
814.	'813'->'0'->'1'->'1100010'->'b'	854.	'853'->'1'->'2'->'1110010'->'r'
815.	'814'->'1'->'1'->'1100001'->'a'	855.	'854'->'1'->'6'->'0100000'->' '
816.	'815'->'1'->'1'->'1100111'->'g'	856.	'855'->'1'->'2'->'1010010'->'R'
817.	'816'->'0'->'2'->'1100001'->'a'	857.	'856'->'1'->'5'->'1110000'->'p'
818.	'817'->'0'->'2'->'1101001'->'i'	858.	'857'->'0'->'2'->'0110101'->'5'
819.	'818'->'1'->'6'->'0100000'->' '	859.	'858'->'0'->'1'->'0111000'->'8'
820.	'819'->'1'->'2'->'1100010'->'b'	860.	'859'->'1'->'2'->'0101110'->','
821.	'820'->'1'->'1'->'1100101'->'e'	861.	'860'->'1'->'1'->'0110111'->'7'
822.	'821'->'0'->'1'->'1110010'->'r'	862.	'861'->'1'->'1'->'0111001'->'9'
823.	'822'->'0'->'2'->'1101001'->'i'	863.	'862'->'1'->'2'->'0110110'->'6'
824.	'823'->'1'->'1'->'1101011'->'k'	864.	'863'->'1'->'2'->'0101110'->','
825.	'824'->'1'->'1'->'1110101'->'u'	865.	'864'->'1'->'4'->'0111000'->'8'
826.	'825'->'1'->'3'->'1110100'->'t'	866.	'865'->'0'->'2'->'0110101'->'5'
827.	'826'->'1'->'2'->'0111010'->','	867.	'866'->'0'->'2'->'0110101'->'5'
828.	'827'->'1'->'3'->'01101'->' '	868.	'867'->'0'->'1'->'0101100'->','
829.	'828'->'1'->'4'->'01010'->' '	869.	'868'->'1'->'5'->'0110000'->'0'
830.	'829'->'0'->'2'->'1100001'->'a'	870.	'869'->'1'->'5'->'0110000'->'0'
831.	'830'->'0'->'1'->'0101110'->','	871.	'870'->'1'->'3'->'01101'->' '
832.	'831'->'0'->'4'->'01001'->'	872.	'871'->'1'->'4'->'01010'->' '
833.	'832'->'1'->'1'->'1000101'->'E'	873.	'872'->'0'->'1'->'1100010'->'b'
834.	'833'->'1'->'1'->'1101101'->'m'	874.	'873'->'0'->'1'->'0101110'->','
835.	'834'->'1'->'2'->'1100010'->'b'	875.	'874'->'1'->'3'->'01001'->'
836.	'835'->'0'->'2'->'1100001'->'a'	876.	'875'->'1'->'1'->'1000101'->'E'
837.	'836'->'1'->'2'->'1110010'->'r'	877.	'876'->'1'->'1'->'1101101'->'m'
838.	'837'->'0'->'3'->'1101011'->'k'	878.	'877'->'1'->'2'->'1100010'->'b'
839.	'838'->'0'->'2'->'1100001'->'a'	879.	'878'->'0'->'2'->'1100001'->'a'
840.	'839'->'0'->'3'->'1110011'->'s'	880.	'879'->'1'->'2'->'1110010'->'r'

881.	'880'->'1'->'1'->'1101011'->'k'	921.	'920'->'0'->'2'->'1101101'->'m'
882.	'881'->'0'->'2'->'1100001'->'a'	922.	'921'->'0'->'1'->'1100010'->'b'
883.	'882'->'1'->'1'->'1110011'->'s'	923.	'922'->'1'->'1'->'1100001'->'a'
884.	'883'->'1'->'1'->'1101001'->'i'	924.	'923'->'0'->'1'->'1110010'->'r'
885.	'884'->'0'->'1'->'0100000'->' '	925.	'924'->'1'->'1'->'1101011'->'k'
886.	'885'->'0'->'2'->'1001101'->'M'	926.	'925'->'1'->'1'->'1100001'->'a'
887.	'886'->'0'->'2'->'1100101'->'e'	927.	'926'->'1'->'1'->'1110011'->'s'
888.	'887'->'0'->'1'->'1100100'->'d'	928.	'927'->'0'->'2'->'1101001'->'i'
889.	'888'->'1'->'1'->'1100001'->'a'	929.	'928'->'0'->'1'->'0100000'->' '
890.	'889'->'1'->'2'->'1101110'->'n'	930.	'929'->'0'->'1'->'1000010'->'B'
891.	'890'->'0'->'4'->'01001'->'	931.	'930'->'1'->'1'->'1100001'->'a'
892.	'891'->'0'->'3'->'1110011'->'s'	932.	'931'->'1'->'3'->'1110100'->'t'
893.	'892'->'0'->'2'->'1100101'->'e'	933.	'932'->'1'->'1'->'1100001'->'a'
894.	'893'->'0'->'1'->'1100010'->'b'	934.	'933'->'0'->'2'->'1101101'->'m'
895.	'894'->'1'->'1'->'1100101'->'e'	935.	'934'->'1'->'3'->'01001'->'
896.	'895'->'0'->'3'->'1110011'->'s'	936.	'935'->'1'->'1'->'1110011'->'s'
897.	'896'->'1'->'1'->'1100001'->'a'	937.	'936'->'0'->'2'->'1100101'->'e'
898.	'897'->'1'->'2'->'1110010'->'r'	938.	'937'->'0'->'1'->'1100010'->'b'
899.	'898'->'1'->'6'->'0100000'->' '	939.	'938'->'1'->'1'->'1100101'->'e'
900.	'899'->'0'->'1'->'1010010'->'R'	940.	'939'->'1'->'1'->'1110011'->'s'
901.	'900'->'1'->'5'->'1110000'->'p'	941.	'940'->'0'->'2'->'1100001'->'a'
902.	'901'->'0'->'2'->'0110101'->'5'	942.	'941'->'0'->'1'->'1110010'->'r'
903.	'902'->'1'->'1'->'0111001'->'9'	943.	'942'->'1'->'6'->'0100000'->' '
904.	'903'->'1'->'2'->'0101110'->'.'	944.	'943'->'0'->'1'->'1010010'->'R'
905.	'904'->'1'->'1'->'0110101'->'5'	945.	'944'->'1'->'5'->'1110000'->'p'
906.	'905'->'0'->'1'->'0110100'->'4'	946.	'945'->'1'->'2'->'0110110'->'6'
907.	'906'->'1'->'1'->'0110111'->'7'	947.	'946'->'1'->'5'->'0110000'->'0'
908.	'907'->'0'->'1'->'0101110'->'.'	948.	'947'->'0'->'1'->'0101110'->'.'
909.	'908'->'1'->'2'->'0110010'->'2'	949.	'948'->'0'->'2'->'0110001'->'1'
910.	'909'->'1'->'2'->'0110010'->'2'	950.	'949'->'1'->'2'->'0110110'->'6'
911.	'910'->'1'->'5'->'0110000'->'0'	951.	'950'->'0'->'3'->'0110011'->'3'
912.	'911'->'1'->'3'->'0101100'->'.'	952.	'951'->'1'->'2'->'0101110'->'.'
913.	'912'->'0'->'1'->'0110000'->'0'	953.	'952'->'1'->'2'->'0110010'->'2'
914.	'913'->'1'->'5'->'0110000'->'0'	954.	'953'->'1'->'1'->'0111001'->'9'
915.	'914'->'0'->'4'->'01101'->' '	955.	'954'->'0'->'2'->'0110101'->'5'
916.	'915'->'1'->'4'->'01010'->' '	956.	'955'->'1'->'3'->'0101100'->'.'
917.	'916'->'1'->'1'->'1100011'->'c'	957.	'956'->'0'->'1'->'0110000'->'0'
918.	'917'->'1'->'2'->'0101110'->'.'	958.	'957'->'1'->'5'->'0110000'->'0'
919.	'918'->'1'->'3'->'01001'->'	959.	'958'->'0'->'4'->'01101'->' '
920.	'919'->'1'->'1'->'1000101'->'E'	960.	'959'->'1'->'4'->'01010'->' '

961.	'960'->'0'->'1'->'1100100'->'d'	1001.	'1000'->'1'->'3'->'0101100'->' '
962.	'961'->'0'->'1'->'0101110'->'.'	1002.	'1001'->'1'->'5'->'0110000'->'0'
963.	'962'->'0'->'4'->'01001'->'	1003.	'1002'->'1'->'5'->'0110000'->'0'
964.	'963'->'0'->'2'->'1000101'->'E'	1004.	'1003'->'0'->'4'->'01101'->' '
965.	'964'->'1'->'1'->'1101101'->'m'	1005.	'1004'->'1'->'4'->'01010'->' '
966.	'965'->'1'->'2'->'1100010'->'b'	1006.	'1005'->'1'->'1'->'1100101'->'e'
967.	'966'->'1'->'1'->'1100001'->'a'	1007.	'1006'->'0'->'1'->'0101110'->'.'
968.	'967'->'1'->'2'->'1110010'->'r'	1008.	'1007'->'0'->'4'->'01001'->'
969.	'968'->'1'->'1'->'1101011'->'k'	1009.	'1008'->'1'->'1'->'1000101'->'E'
970.	'969'->'0'->'2'->'1100001'->'a'	1010.	'1009'->'1'->'1'->'1101101'->'m'
971.	'970'->'0'->'3'->'1110011'->'s'	1011.	'1010'->'0'->'1'->'1100010'->'b'
972.	'971'->'1'->'1'->'1101001'->'i'	1012.	'1011'->'0'->'2'->'1100001'->'a'
973.	'972'->'0'->'1'->'0100000'->' '	1013.	'1012'->'0'->'1'->'1110010'->'r'
974.	'973'->'1'->'5'->'1010000'->'P'	1014.	'1013'->'1'->'1'->'1101011'->'k'
975.	'974'->'1'->'1'->'1100001'->'a'	1015.	'1014'->'0'->'2'->'1100001'->'a'
976.	'975'->'1'->'3'->'1100100'->'d'	1016.	'1015'->'1'->'1'->'1110011'->'s'
977.	'976'->'0'->'2'->'1100001'->'a'	1017.	'1016'->'1'->'1'->'1101001'->'i'
978.	'977'->'1'->'2'->'1101110'->'n'	1018.	'1017'->'1'->'6'->'0100000'->' '
979.	'978'->'0'->'4'->'1100111'->'g'	1019.	'1018'->'0'->'1'->'1010000'->'P'
980.	'979'->'1'->'3'->'01001'->'	1020.	'1019'->'0'->'2'->'1100001'->'a'
981.	'980'->'1'->'1'->'1110011'->'s'	1021.	'1020'->'0'->'1'->'1101100'->'i'
982.	'981'->'1'->'1'->'1100101'->'e'	1022.	'1021'->'1'->'1'->'1100101'->'e'
983.	'982'->'0'->'1'->'1100010'->'b'	1023.	'1022'->'1'->'1'->'1101101'->'m'
984.	'983'->'1'->'1'->'1100101'->'e'	1024.	'1023'->'0'->'1'->'1100010'->'b'
985.	'984'->'0'->'3'->'1110011'->'s'	1025.	'1024'->'0'->'2'->'1100001'->'a'
986.	'985'->'0'->'2'->'1100001'->'a'	1026.	'1025'->'1'->'2'->'1101110'->'n'
987.	'986'->'1'->'2'->'1110010'->'r'	1027.	'1026'->'1'->'1'->'1100111'->'g'
988.	'987'->'1'->'6'->'0100000'->' '	1028.	'1027'->'1'->'3'->'01001'->'
989.	'988'->'1'->'2'->'1010010'->'R'	1029.	'1028'->'0'->'3'->'1110011'->'s'
990.	'989'->'0'->'1'->'1110000'->'p'	1030.	'1029'->'1'->'1'->'1100101'->'e'
991.	'990'->'0'->'1'->'0110110'->'6'	1031.	'1030'->'1'->'2'->'1100010'->'b'
992.	'991'->'1'->'5'->'0110000'->'0'	1032.	'1031'->'1'->'1'->'1100101'->'e'
993.	'992'->'1'->'2'->'0101110'->'.'	1033.	'1032'->'0'->'3'->'1110011'->'s'
994.	'993'->'1'->'1'->'0110111'->'7'	1034.	'1033'->'0'->'2'->'1100001'->'a'
995.	'994'->'1'->'1'->'0110111'->'7'	1035.	'1034'->'0'->'1'->'1110010'->'r'
996.	'995'->'1'->'1'->'0110101'->'5'	1036.	'1035'->'0'->'1'->'0100000'->' '
997.	'996'->'0'->'1'->'0101110'->'.'	1037.	'1036'->'1'->'2'->'1010010'->'R'
998.	'997'->'0'->'1'->'0110000'->'0'	1038.	'1037'->'1'->'5'->'1110000'->'p'
999.	'998'->'0'->'2'->'0111001'->'9'	1039.	'1038'->'1'->'2'->'0110110'->'6'
1000.	'999'->'1'->'5'->'0110000'->'0'	1040.	'1039'->'1'->'1'->'0110001'->'1'



1041.	'1040'->'0'->'1'->'0101110'->.'	1081.	'1080'->'0'->'3'->'1101011'->'k'
1042.	'1041'->'0'->'1'->'0110010'->'2'	1082.	'1081'->'1'->'6'->'0100000'->' '
1043.	'1042'->'1'->'1'->'0110011'->'3'	1083.	'1082'->'0'->'4'->'1000111'->'G'
1044.	'1043'->'1'->'2'->'0110110'->'6'	1084.	'1083'->'0'->'2'->'1100101'->'e'
1045.	'1044'->'1'->'2'->'0101110'->.'	1085.	'1084'->'1'->'3'->'1100100'->'d'
1046.	'1045'->'1'->'1'->'0110101'->'5'	1086.	'1085'->'1'->'1'->'1100101'->'e'
1047.	'1046'->'0'->'1'->'0110010'->'2'	1087.	'1086'->'1'->'1'->'0101001'->')'
1048.	'1047'->'0'->'1'->'0110000'->'0'	1088.	'1087'->'1'->'3'->'01001'->'
1049.	'1048'->'0'->'1'->'0101100'->','	1089.	'1088'->'1'->'1'->'1110011'->'s'
1050.	'1049'->'1'->'5'->'0110000'->'0'	1090.	'1089'->'0'->'2'->'1100101'->'e'
1051.	'1050'->'1'->'5'->'0110000'->'0'	1091.	'1090'->'0'->'1'->'1100010'->'b'
1052.	'1051'->'0'->'4'->'01101'->' '	1092.	'1091'->'1'->'1'->'1100101'->'e'
1053.	'1052'->'0'->'3'->'01010'->' '	1093.	'1092'->'1'->'1'->'1110011'->'s'
1054.	'1053'->'1'->'2'->'1100110'->'f'	1094.	'1093'->'1'->'1'->'1100001'->'a'
1055.	'1054'->'0'->'1'->'0101110'->.'	1095.	'1094'->'0'->'1'->'1110010'->'r'
1056.	'1055'->'1'->'3'->'01001'->'	1096.	'1095'->'1'->'6'->'0100000'->' '
1057.	'1056'->'1'->'1'->'1000101'->'E'	1097.	'1096'->'1'->'2'->'1010010'->'R'
1058.	'1057'->'1'->'1'->'1101101'->'m'	1098.	'1097'->'1'->'5'->'1110000'->'p'
1059.	'1058'->'1'->'2'->'1100010'->'b'	1099.	'1098'->'1'->'2'->'0110110'->'6'
1060.	'1059'->'0'->'2'->'1100001'->'a'	1100.	'1099'->'1'->'2'->'0110010'->'2'
1061.	'1060'->'1'->'2'->'1110010'->'r'	1101.	'1100'->'1'->'2'->'0101110'->.'
1062.	'1061'->'0'->'3'->'1101011'->'k'	1102.	'1101'->'1'->'2'->'0110010'->'2'
1063.	'1062'->'0'->'2'->'1100001'->'a'	1103.	'1102'->'0'->'3'->'0110011'->'3'
1064.	'1063'->'0'->'3'->'1110011'->'s'	1104.	'1103'->'1'->'1'->'0111001'->'9'
1065.	'1064'->'1'->'1'->'1101001'->'i'	1105.	'1104'->'0'->'1'->'0101110'->.'
1066.	'1065'->'1'->'6'->'0100000'->' '	1106.	'1105'->'1'->'5'->'0110000'->'0'
1067.	'1066'->'1'->'2'->'1001010'->'J'	1107.	'1106'->'1'->'1'->'0110011'->'3'
1068.	'1067'->'1'->'1'->'1100001'->'a'	1108.	'1107'->'1'->'1'->'0110101'->'5'
1069.	'1068'->'0'->'3'->'1101011'->'k'	1109.	'1108'->'0'->'1'->'0101100'->','
1070.	'1069'->'0'->'2'->'1100001'->'a'	1110.	'1109'->'1'->'5'->'0110000'->'0'
1071.	'1070'->'1'->'2'->'1110010'->'r'	1111.	'1110'->'1'->'5'->'0110000'->'0'
1072.	'1071'->'1'->'3'->'1110100'->'t'	1112.	'1111'->'0'->'4'->'01101'->' '
1073.	'1072'->'1'->'1'->'1100001'->'a'	1113.	'1112'->'1'->'4'->'01010'->' '
1074.	'1073'->'0'->'1'->'0100000'->' '	1114.	'1113'->'1'->'1'->'1100111'->'g'
1075.	'1074'->'0'->'1'->'0101000'->'('	1115.	'1114'->'0'->'1'->'0101110'->.'
1076.	'1075'->'1'->'5'->'1010000'->'P'	1116.	'1115'->'1'->'3'->'01001'->'
1077.	'1076'->'1'->'1'->'1101111'->'o'	1117.	'1116'->'0'->'2'->'1000101'->'E'
1078.	'1077'->'1'->'2'->'1101110'->'n'	1118.	'1117'->'0'->'2'->'1101101'->'m'
1079.	'1078'->'1'->'3'->'1100100'->'d'	1119.	'1118'->'0'->'1'->'1100010'->'b'
1080.	'1079'->'1'->'1'->'1101111'->'o'	1120.	'1119'->'1'->'1'->'1100001'->'a'

1121.	'1120'->'1'->'2'->'1110010'->'r'	1161.	'1160'->'1'->'5'->'0110000'->'0'
1122.	'1121'->'1'->'1'->'1101011'->'k'	1162.	'1161'->'0'->'3'->'0110011'->'3'
1123.	'1122'->'1'->'1'->'1100001'->'a'	1163.	'1162'->'1'->'1'->'0110101'->'5'
1124.	'1123'->'0'->'3'->'1110011'->'s'	1164.	'1163'->'0'->'1'->'0101100'->','
1125.	'1124'->'1'->'1'->'1101001'->'i'	1165.	'1164'->'0'->'1'->'0110000'->'0'
1126.	'1125'->'0'->'1'->'0100000'->' '	1166.	'1165'->'0'->'1'->'0110000'->'0'
1127.	'1126'->'0'->'1'->'1001010'->'j'	1167.	'1166'->'1'->'3'->'01101'->' '
1128.	'1127'->'1'->'1'->'1100001'->'a'	1168.	'1167'->'1'->'4'->'01010'->' '
1129.	'1128'->'1'->'1'->'1101011'->'k'	1169.	'1168'->'0'->'1'->'1101000'->'h'
1130.	'1129'->'1'->'1'->'1100001'->'a'	1170.	'1169'->'1'->'2'->'0101110'->'.'
1131.	'1130'->'0'->'1'->'1110010'->'r'	1171.	'1170'->'1'->'3'->'01001'->'
1132.	'1131'->'1'->'3'->'1110100'->'t'	1172.	'1171'->'0'->'2'->'1000101'->'E'
1133.	'1132'->'0'->'2'->'1100001'->'a'	1173.	'1172'->'0'->'2'->'1101101'->'m'
1134.	'1133'->'1'->'6'->'0100000'->' '	1174.	'1173'->'1'->'2'->'1100010'->'b'
1135.	'1134'->'1'->'4'->'0101000'->'('	1175.	'1174'->'1'->'1'->'1100001'->'a'
1136.	'1135'->'1'->'2'->'1000010'->'B'	1176.	'1175'->'1'->'2'->'1110010'->'r'
1137.	'1136'->'1'->'1'->'1100101'->'e'	1177.	'1176'->'1'->'1'->'1101011'->'k'
1138.	'1137'->'1'->'1'->'1101011'->'k'	1178.	'1177'->'1'->'1'->'1100001'->'a'
1139.	'1138'->'0'->'2'->'1100001'->'a'	1179.	'1178'->'1'->'1'->'1110011'->'s'
1140.	'1139'->'0'->'3'->'1110011'->'s'	1180.	'1179'->'0'->'2'->'1101001'->'i'
1141.	'1140'->'1'->'1'->'1101001'->'i'	1181.	'1180'->'0'->'1'->'0100000'->' '
1142.	'1141'->'1'->'1'->'0101001'->')'	1182.	'1181'->'0'->'3'->'1010011'->'S'
1143.	'1142'->'1'->'3'->'01001'->'	1183.	'1182'->'0'->'5'->'1101111'->'o'
1144.	'1143'->'1'->'1'->'1110011'->'s'	1184.	'1183'->'1'->'3'->'1101100'->'l'
1145.	'1144'->'0'->'2'->'1100101'->'e'	1185.	'1184'->'0'->'5'->'1101111'->'o'
1146.	'1145'->'0'->'1'->'1100010'->'b'	1186.	'1185'->'0'->'4'->'01001'->'
1147.	'1146'->'0'->'2'->'1100101'->'e'	1187.	'1186'->'1'->'1'->'1110011'->'s'
1148.	'1147'->'0'->'3'->'1110011'->'s'	1188.	'1187'->'0'->'2'->'1100101'->'e'
1149.	'1148'->'1'->'1'->'1100001'->'a'	1189.	'1188'->'0'->'1'->'1100010'->'b'
1150.	'1149'->'1'->'2'->'1110010'->'r'	1190.	'1189'->'0'->'2'->'1100101'->'e'
1151.	'1150'->'0'->'1'->'0100000'->' '	1191.	'1190'->'1'->'1'->'1110011'->'s'
1152.	'1151'->'1'->'2'->'1010010'->'R'	1192.	'1191'->'1'->'1'->'1100001'->'a'
1153.	'1152'->'1'->'5'->'1110000'->'p'	1193.	'1192'->'1'->'2'->'1110010'->'r'
1154.	'1153'->'0'->'1'->'0110110'->'6'	1194.	'1193'->'0'->'1'->'0100000'->' '
1155.	'1154'->'1'->'2'->'0110010'->'2'	1195.	'1194'->'1'->'2'->'1010010'->'R'
1156.	'1155'->'1'->'2'->'0101110'->'.'	1196.	'1195'->'0'->'1'->'1110000'->'p'
1157.	'1156'->'1'->'2'->'0110010'->'2'	1197.	'1196'->'0'->'1'->'0110110'->'6'
1158.	'1157'->'0'->'3'->'0110011'->'3'	1198.	'1197'->'1'->'1'->'0110011'->'3'
1159.	'1158'->'1'->'1'->'0111001'->'9'	1199.	'1198'->'1'->'2'->'0101110'->'.'
1160.	'1159'->'0'->'1'->'0101110'->'.'	1200.	'1199'->'0'->'1'->'0110110'->'6'

1201.	'1200'->'0'->'1'->'0110100'->'4'	1241.	'1240'->'1'->'6'->'0100000'->' '
1202.	'1201'->'1'->'5'->'0110000'->'0'	1242.	'1241'->'1'->'2'->'1010010'->'R'
1203.	'1202'->'1'->'2'->'0101110'->'.'	1243.	'1242'->'0'->'1'->'1110000'->'p'
1204.	'1203'->'0'->'2'->'0110001'->'1'	1244.	'1243'->'1'->'2'->'0110110'->'6'
1205.	'1204'->'1'->'2'->'0110010'->'2'	1245.	'1244'->'0'->'3'->'0110011'->'3'
1206.	'1205'->'1'->'5'->'0110000'->'0'	1246.	'1245'->'1'->'2'->'0101110'->'.'
1207.	'1206'->'1'->'3'->'0101100'->','	1247.	'1246'->'1'->'1'->'0110111'->'7'
1208.	'1207'->'0'->'1'->'0110000'->'0'	1248.	'1247'->'1'->'1'->'0111001'->'9'
1209.	'1208'->'0'->'1'->'0110000'->'0'	1249.	'1248'->'1'->'4'->'0111000'->'8'
1210.	'1209'->'1'->'3'->'01101'->' '	1250.	'1249'->'0'->'1'->'0101110'->'.'
1211.	'1210'->'1'->'4'->'01010'->' '	1251.	'1250'->'1'->'2'->'0110110'->'6'
1212.	'1211'->'1'->'1'->'1101001'->'i'	1252.	'1251'->'0'->'2'->'0111001'->'9'
1213.	'1212'->'1'->'2'->'0101110'->'.'	1253.	'1252'->'0'->'1'->'0110000'->'0'
1214.	'1213'->'0'->'4'->'01001'->'	1254.	'1253'->'1'->'3'->'0101100'->','
1215.	'1214'->'1'->'1'->'1000101'->'E'	1255.	'1254'->'1'->'5'->'0110000'->'0'
1216.	'1215'->'1'->'1'->'1101101'->'m'	1256.	'1255'->'1'->'5'->'0110000'->'0'
1217.	'1216'->'0'->'1'->'1100010'->'b'	1257.	'1256'->'1'->'3'->'01101'->' '
1218.	'1217'->'1'->'1'->'1100001'->'a'	1258.	'1257'->'0'->'3'->'01010'->' '
1219.	'1218'->'1'->'2'->'1110010'->'r'	1259.	'1258'->'1'->'2'->'1101010'->'j'
1220.	'1219'->'1'->'1'->'1101011'->'k'	1260.	'1259'->'0'->'1'->'0101110'->'.'
1221.	'1220'->'0'->'2'->'1100001'->'a'	1261.	'1260'->'1'->'3'->'01001'->'
1222.	'1221'->'1'->'1'->'1110011'->'s'	1262.	'1261'->'1'->'1'->'1000101'->'E'
1223.	'1222'->'0'->'2'->'1101001'->'i'	1263.	'1262'->'1'->'1'->'1101101'->'m'
1224.	'1223'->'1'->'6'->'0100000'->' '	1264.	'1263'->'1'->'2'->'1100010'->'b'
1225.	'1224'->'0'->'3'->'1010011'->'S'	1265.	'1264'->'0'->'2'->'1100001'->'a'
1226.	'1225'->'1'->'1'->'1110101'->'u'	1266.	'1265'->'0'->'1'->'1110010'->'r'
1227.	'1226'->'1'->'2'->'1110010'->'r'	1267.	'1266'->'1'->'1'->'1101011'->'k'
1228.	'1227'->'1'->'1'->'1100001'->'a'	1268.	'1267'->'1'->'1'->'1100001'->'a'
1229.	'1228'->'0'->'1'->'1100010'->'b'	1269.	'1268'->'1'->'1'->'1110011'->'s'
1230.	'1229'->'0'->'2'->'1100001'->'a'	1270.	'1269'->'1'->'1'->'1101001'->'i'
1231.	'1230'->'0'->'2'->'1111001'->'y'	1271.	'1270'->'0'->'1'->'0100000'->' '
1232.	'1231'->'0'->'2'->'1100001'->'a'	1272.	'1271'->'1'->'2'->'1000010'->'B'
1233.	'1232'->'1'->'3'->'01001'->'	1273.	'1272'->'1'->'1'->'1100001'->'a'
1234.	'1233'->'1'->'1'->'1110011'->'s'	1274.	'1273'->'1'->'2'->'1101110'->'n'
1235.	'1234'->'0'->'2'->'1100101'->'e'	1275.	'1274'->'1'->'2'->'1101010'->'j'
1236.	'1235'->'0'->'1'->'1100010'->'b'	1276.	'1275'->'1'->'1'->'1100001'->'a'
1237.	'1236'->'0'->'2'->'1100101'->'e'	1277.	'1276'->'0'->'1'->'1110010'->'r'
1238.	'1237'->'1'->'1'->'1110011'->'s'	1278.	'1277'->'1'->'1'->'1101101'->'m'
1239.	'1238'->'1'->'1'->'1100001'->'a'	1279.	'1278'->'1'->'1'->'1100001'->'a'
1240.	'1239'->'1'->'2'->'1110010'->'r'	1280.	'1279'->'1'->'1'->'1110011'->'s'

1281.	'1280'->'0'->'2'->'1101001'->'i'	1321.	'1320'->'0'->'1'->'0100000'->' '
1282.	'1281'->'1'->'2'->'1101110'->'n'	1322.	'1321'->'1'->'2'->'1000010'->'B'
1283.	'1282'->'0'->'4'->'01001'->'	1323.	'1322'->'1'->'1'->'1100001'->'a'
1284.	'1283'->'0'->'3'->'1110011'->'s'	1324.	'1323'->'1'->'3'->'1101100'->'l'
1285.	'1284'->'0'->'2'->'1100101'->'e'	1325.	'1324'->'1'->'1'->'1101001'->'i'
1286.	'1285'->'1'->'2'->'1100010'->'b'	1326.	'1325'->'0'->'3'->'1101011'->'k'
1287.	'1286'->'0'->'2'->'1100101'->'e'	1327.	'1326'->'1'->'5'->'1110000'->'p'
1288.	'1287'->'1'->'1'->'1110011'->'s'	1328.	'1327'->'0'->'2'->'1100001'->'a'
1289.	'1288'->'1'->'1'->'1100001'->'a'	1329.	'1328'->'1'->'5'->'1110000'->'p'
1290.	'1289'->'1'->'2'->'1110010'->'r'	1330.	'1329'->'1'->'1'->'1100001'->'a'
1291.	'1290'->'1'->'6'->'0100000'->' '	1331.	'1330'->'1'->'2'->'1101110'->'n'
1292.	'1291'->'0'->'1'->'1010010'->'R'	1332.	'1331'->'1'->'3'->'01001'->'
1293.	'1292'->'1'->'5'->'1110000'->'p'	1333.	'1332'->'0'->'3'->'1110011'->'s'
1294.	'1293'->'1'->'2'->'0110110'->'6'	1334.	'1333'->'1'->'1'->'1100101'->'e'
1295.	'1294'->'1'->'1'->'0110101'->'5'	1335.	'1334'->'0'->'1'->'1100010'->'b'
1296.	'1295'->'1'->'2'->'0101110'->'.'	1336.	'1335'->'1'->'1'->'1100101'->'e'
1297.	'1296'->'1'->'4'->'0111000'->'8'	1337.	'1336'->'0'->'3'->'1110011'->'s'
1298.	'1297'->'1'->'2'->'0110110'->'6'	1338.	'1337'->'1'->'1'->'1100001'->'a'
1299.	'1298'->'0'->'3'->'0110011'->'3'	1339.	'1338'->'1'->'2'->'1110010'->'r'
1300.	'1299'->'1'->'2'->'0101110'->'.'	1340.	'1339'->'0'->'1'->'0100000'->' '
1301.	'1300'->'1'->'1'->'0111001'->'9'	1341.	'1340'->'0'->'1'->'1010010'->'R'
1302.	'1301'->'1'->'2'->'0110010'->'2'	1342.	'1341'->'0'->'1'->'1110000'->'p'
1303.	'1302'->'1'->'1'->'0111001'->'9'	1343.	'1342'->'1'->'2'->'0110110'->'6'
1304.	'1303'->'1'->'3'->'0101100'->'.'	1344.	'1343'->'1'->'2'->'0110110'->'6'
1305.	'1304'->'0'->'1'->'0110000'->'0'	1345.	'1344'->'1'->'2'->'0101110'->'.'
1306.	'1305'->'1'->'5'->'0110000'->'0'	1346.	'1345'->'1'->'2'->'0110010'->'2'
1307.	'1306'->'1'->'3'->'01101'->' '	1347.	'1346'->'1'->'1'->'0110011'->'3'
1308.	'1307'->'1'->'4'->'01010'->' '	1348.	'1347'->'0'->'1'->'0110010'->'2'
1309.	'1308'->'1'->'1'->'1101011'->'k'	1349.	'1348'->'1'->'2'->'0101110'->'.'
1310.	'1309'->'1'->'2'->'0101110'->'.'	1350.	'1349'->'1'->'2'->'0110010'->'2'
1311.	'1310'->'1'->'3'->'01001'->'	1351.	'1350'->'0'->'2'->'0111001'->'9'
1312.	'1311'->'0'->'2'->'1000101'->'E'	1352.	'1351'->'1'->'5'->'0110000'->'0'
1313.	'1312'->'0'->'2'->'1101101'->'m'	1353.	'1352'->'1'->'3'->'0101100'->'.'
1314.	'1313'->'1'->'2'->'1100010'->'b'	1354.	'1353'->'0'->'1'->'0110000'->'0'
1315.	'1314'->'1'->'1'->'1100001'->'a'	1355.	'1354'->'1'->'5'->'0110000'->'0'
1316.	'1315'->'1'->'2'->'1110010'->'r'	1356.	'1355'->'0'->'4'->'01101'->' '
1317.	'1316'->'1'->'1'->'1101011'->'k'	1357.	'1356'->'1'->'4'->'01010'->' '
1318.	'1317'->'1'->'1'->'1100001'->'a'	1358.	'1357'->'1'->'3'->'1101100'->'l'
1319.	'1318'->'1'->'1'->'1110011'->'s'	1359.	'1358'->'1'->'2'->'0101110'->'.'
1320.	'1319'->'0'->'2'->'1101001'->'i'	1360.	'1359'->'1'->'3'->'01001'->'

1361.	'1360'->'1'->'1'->'1000101'->'E'	1401.	'1400'->'1'->'5'->'0110000'->'0'
1362.	'1361'->'1'->'1'->'1101101'->'m'	1402.	'1401'->'1'->'5'->'0110000'->'0'
1363.	'1362'->'0'->'1'->'1100010'->'b'	1403.	'1402'->'0'->'4'->'01101'->' '
1364.	'1363'->'0'->'2'->'1100001'->'a'	1404.	'1403'->'0'->'3'->'01010'->' '
1365.	'1364'->'1'->'2'->'1110010'->'r'	1405.	'1404'->'1'->'1'->'1101101'->'m'
1366.	'1365'->'1'->'1'->'1101011'->'k'	1406.	'1405'->'0'->'1'->'0101110'->'.'
1367.	'1366'->'0'->'2'->'1100001'->'a'	1407.	'1406'->'0'->'4'->'01001'->'
1368.	'1367'->'1'->'1'->'1110011'->'s'	1408.	'1407'->'1'->'1'->'1000101'->'E'
1369.	'1368'->'0'->'2'->'1101001'->'i'	1409.	'1408'->'0'->'2'->'1101101'->'m'
1370.	'1369'->'0'->'1'->'0100000'->' '	1410.	'1409'->'1'->'2'->'1100010'->'b'
1371.	'1370'->'1'->'1'->'1001101'->'M'	1411.	'1410'->'0'->'2'->'1100001'->'a'
1372.	'1371'->'1'->'1'->'1100001'->'a'	1412.	'1411'->'1'->'2'->'1110010'->'r'
1373.	'1372'->'1'->'1'->'1101011'->'k'	1413.	'1412'->'0'->'3'->'1101011'->'k'
1374.	'1373'->'1'->'1'->'1100001'->'a'	1414.	'1413'->'0'->'2'->'1100001'->'a'
1375.	'1374'->'0'->'3'->'1110011'->'s'	1415.	'1414'->'1'->'1'->'1110011'->'s'
1376.	'1375'->'0'->'3'->'1110011'->'s'	1416.	'1415'->'1'->'1'->'1101001'->'i'
1377.	'1376'->'0'->'2'->'1100001'->'a'	1417.	'1416'->'0'->'1'->'0100000'->' '
1378.	'1377'->'1'->'2'->'1110010'->'r'	1418.	'1417'->'1'->'3'->'1001100'->'L'
1379.	'1378'->'0'->'4'->'01001'->'	1419.	'1418'->'0'->'5'->'1101111'->'o'
1380.	'1379'->'1'->'1'->'1110011'->'s'	1420.	'1419'->'1'->'1'->'1101101'->'m'
1381.	'1380'->'1'->'1'->'1100101'->'e'	1421.	'1420'->'0'->'1'->'1100010'->'b'
1382.	'1381'->'1'->'2'->'1100010'->'b'	1422.	'1421'->'1'->'1'->'1101111'->'o'
1383.	'1382'->'0'->'2'->'1100101'->'e'	1423.	'1422'->'1'->'1'->'1101011'->'k'
1384.	'1383'->'1'->'1'->'1110011'->'s'	1424.	'1423'->'0'->'4'->'01001'->'
1385.	'1384'->'0'->'2'->'1100001'->'a'	1425.	'1424'->'1'->'1'->'1110011'->'s'
1386.	'1385'->'1'->'2'->'1110010'->'r'	1426.	'1425'->'0'->'2'->'1100101'->'e'
1387.	'1386'->'1'->'6'->'0100000'->' '	1427.	'1426'->'0'->'1'->'1100010'->'b'
1388.	'1387'->'1'->'2'->'1010010'->'R'	1428.	'1427'->'0'->'2'->'1100101'->'e'
1389.	'1388'->'0'->'1'->'1110000'->'p'	1429.	'1428'->'1'->'1'->'1110011'->'s'
1390.	'1389'->'1'->'2'->'0110110'->'6'	1430.	'1429'->'1'->'1'->'1100001'->'a'
1391.	'1390'->'0'->'4'->'0110111'->'7'	1431.	'1430'->'0'->'1'->'1110010'->'r'
1392.	'1391'->'1'->'2'->'0101110'->'.'	1432.	'1431'->'0'->'1'->'0100000'->' '
1393.	'1392'->'0'->'1'->'0110010'->'2'	1433.	'1432'->'0'->'1'->'1010010'->'R'
1394.	'1393'->'1'->'1'->'0110001'->'1'	1434.	'1433'->'1'->'5'->'1110000'->'p'
1395.	'1394'->'1'->'3'->'0110100'->'4'	1435.	'1434'->'0'->'1'->'0110110'->'6'
1396.	'1395'->'1'->'2'->'0101110'->'.'	1436.	'1435'->'1'->'2'->'0110110'->'6'
1397.	'1396'->'0'->'2'->'0110101'->'5'	1437.	'1436'->'1'->'2'->'0101110'->'.'
1398.	'1397'->'0'->'1'->'0111000'->'8'	1438.	'1437'->'1'->'1'->'0110101'->'5'
1399.	'1398'->'0'->'1'->'0110110'->'6'	1439.	'1438'->'0'->'1'->'0110000'->'0'
1400.	'1399'->'0'->'1'->'0101100'->','	1440.	'1439'->'1'->'1'->'0110101'->'5'

1441.	'1440'->'1'->'2'->'0101110'->'.'
1442.	'1441'->'1'->'1'->'0110001'->'1'
1443.	'1442'->'1'->'1'->'0110101'->'5'
1444.	'1443'->'1'->'5'->'0110000'->'0'
1445.	'1444'->'0'->'1'->'0101100'->','
1446.	'1445'->'1'->'5'->'0110000'->'0'
1447.	'1446'->'0'->'1'->'0110000'->'0'
1448.	'1447'->'1'->'3'->'01101'->' '
1449.	'1448'->'0'->'3'->'01010'->' '

## BIOGRAFI PENULIS



**Alfin Naharuddin**, Magister di Institut Teknologi Sepuluh Nopember Surabaya, Fakultas Teknologi Elektro, Departemen Teknik Elektro, Bidang Keahlian Telematika – Pengelola Teknologi Informasi dan Komunikasi (PeTIK) dengan beasiswa dari Kementerian Komunikasi dan Informasi RI. angkatan Tahun 2016. Lahir di Sidoarjo pada 11 Mei 1980, anak kedua dari tiga bersaudara dari pasangan Bapak Akhmad Zarnuji (alm.) dengan Ibu Kholifah, telah menikah dengan Risna Sahara dan telah dikarunia dua orang putera-puteri penyejuk hati Ahmad Azka Naharuddin dan Ayataqiya Naharuddin. Pada saat ini bertugas di Kantor Wilayah Kementerian Agama Provinsi Aceh sebagai Pranata Komputer Muda dan berdomisili di Banda Aceh dengan alamat email [alfin.nahr@gmail.com](mailto:alfin.nahr@gmail.com).

### Riwayat Pendidikan

- 1987 – 1993 MI Faqih Hasyim Sidoarjo
- 1993 – 1996 MTsN Sidoarjo
- 1996 – 1999 MAK N Jember
- 2001 – 2005 Universitas Muhammadiyah Sidoarjo

### Riwayat Pekerjaan

- 2000 – 2007 Staff HRD PT Japfa Comfeed Indonesia, Tbk. Sidoarjo
- 2008 – 2012 Pengelola Kepegawaian Subbag Ortala & Kepegawaian pada Kantor Wilayah Kementerian Agama Provinsi Aceh
- 2012 – 2014 Pranata Komputer Pertama pada Kantor Wilayah Kementerian Agama Provinsi Aceh
- 2014 – 2018 Pranata Komputer Muda pada Kantor Wilayah Kementerian Agama Provinsi Aceh